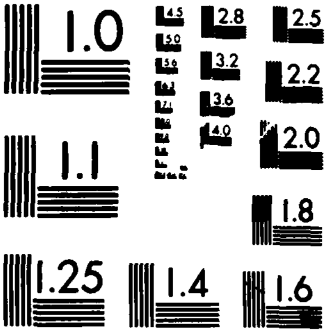
AD-A179 964   2-D BISTABLE OPTICAL CIRCUITS(U) MASSACHUSETTS INST OF   1/1
TECH CAMBRIDGE DEPT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE C WARDE 30 JAN 87 AFOSR-TR-87-0420
UNCLASSIFIED   AFOSR-84-0358   F/G 9/5   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2-D BISTABLE OPTICAL CIRCUITS

FINAL REPORT REPORT, AFOSR-84-0358

30 January , 1987

Cardinal Warde

Department of Electrical Engineering
and Computer Science

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Bolling Air Force Base, D.C. 20332

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
UNLIMITED

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 87-0420

6a. NAME OF PERFORMING ORGANIZATION
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

6c. ADDRESS *(City, State, and ZIP Code)*
17 MASSACHUSETTS AVE.
CAMBRIDGE, MA 02139

7b. ADDRESS *(City, State, and ZIP Code)*
BOLLING AIR FORCE BASE / Bldg 410
WASHINGTON, D. C. 20332

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Same as 7A

8b. OFFICE SYMBOL *(If applicable)*
NE

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR - 84 - 0358

8c. ADDRESS *(City, State, and ZIP Code)*
Same as 7B

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61102F	2305	B4	

11. TITLE *(Include Security Classification)*
2D BISTABLE OPTICAL CIRCUITS

12. PERSONAL AUTHOR(S)
WARDE, CARDINAL

13a. TYPE OF REPORT
FINAL

13b. TIME COVERED
FROM 1/9/84 TO 30/11/85

14. DATE OF REPORT *(Year, Month, Day)*
30 JAN 87

15. PAGE COUNT
42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
BISTABILITY, SPATIAL LIGHT MODULATORS, OPTICAL INFORMATION PROCESSING, FEEDBACK.

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

two dimensional

This final report summarizes an ~~fifteen-month~~ investigation of: (1) 2-D arrays of bistable optical elements and circuits based on microchannel spatial light modulator (MSLM) technology; (2) an investigation of the possible applications of the balanced ternary system (BTS) in digital optical processing and (3) a preliminary study of the design of hybrid optical inference machines. The bistable circuit arrays employ optical feedback around an MSLM, and a novel device called the beamlet array generator which can generate a discrete 2D array of millions of laser beamlets from a single input laser beam. A general purpose versatile circuit is duscussed as well as systems for phase-driven bistability and amplitude-driven bistability. In the BTS study, it is shown that the BTS offers several advantages over the conventional binary system. These include more efficient use of optical power and space-bandwidth product coupled with increased dynamic range when performing multiplication. Finally, in the symbolic processing program, two novel hybrid optical architectures are proposed for parallel symbolic inference on electronic knowledge bases. Keywords:

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Carter

22b. TELEPHONE *(Include Area Code)*
767-4931

22c. OFFICE SYMBOL
NE

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

2-D BISTABLE OPTICAL CIRCUITS

FINAL REPORT REPORT, AFOSR-84-0358

30 January , 1987

Cardinal Warde

Department of Electrical Engineering
and Computer Science

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

Approved for Public Release
Distribution Unlimited

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Bolling Air Force Base, D.C. 20332

TABLE OF CONTENTS

LIST OF FIGURES

SECTION I - 2-D BISTABLE OPTICAL CIRCUITS

INTRODUCTION

The development of a high-resolution, two-dimensional array of optical bistable circuits that offer low switching threshold intensity and high framing rate would be of significant benefit to the fields of digital optical processing, neural network processing, optical computer and VLSI interconnections and for general purpose optical information processing. Such bistable arrays could be used to perform 2-D logic operations, 2-D latches for digital memory, level restoration, level amplification, 2-D analog-to-digital conversion, thresholding, real-time image half toning, sample and hold operations, and 2-D optically-controlled switches.

The desired input-output characteristic of our optical bistable circuits is illustrated in Fig. 1. Because of two recent important device developments in our laboratory, it is now possible to realize programmable 2-D arrays of bistable optical circuits as well as several classes of versatile, all-optical, open and closed loop circuits. The two key devices are: the laser beamlet array generator (BAG) and the microchannel spatial light modulator (MSLM).

MSLM

The MSLM is illustrated in Fig. 2 and is described in detail elsewhere.[1-3] Essentially, it consists of a sandwich of photocathode material, a microchannel plate (MCP), an acceleration grid and a thin electro-optic crystal. The electro-optic crystal is coated on the side facing the MCP with a dielectric mirror and on the other with a transparent electrode. This entire assembly is enclosed in an evacuated tube. The important and unique characteristics of the MSLM that make it potentially well suited for these optical circuits are its low halfwave exposure (~1-30 nJ/cm^2), high framing speed (>50 Hz), its level thresholding and image storage characteristics, and the ability to write and readout the device with light of the same wavelength (cascadability). The latter characteristic is of utmost importance because it permits the realization of cascadable optical circuits.

BEAMLET ARRAY GENERATOR

The principle of the laser beamlet array generator is illustrated in Fig. 3(a). It is essentially two cascaded Michelson interferometers. This novel device can generate a 2-D matrix of millions of regularly spaced laser beamlets on adjustable center-to-center spacing from a single input laser beam. The beamlets remain well resolved both in the near and far field. The most recent improvement of this device uses a special prism instead of two beamsplitters (See Fig. 3b). The performance of the device can be improved even further by optimizing the temperature coefficient of the device and suppressing the remaining multiple reflections which lead to "ghost" beams. These can be accomplished by using quartz rather than BK7 glasss for the prism, and immersing the entire system in refractive index matching oil. A sample of the beams from a prototype beamlet array generator is shown in Fig. 4(a). These beamlets have been used to sample images as illustrated in

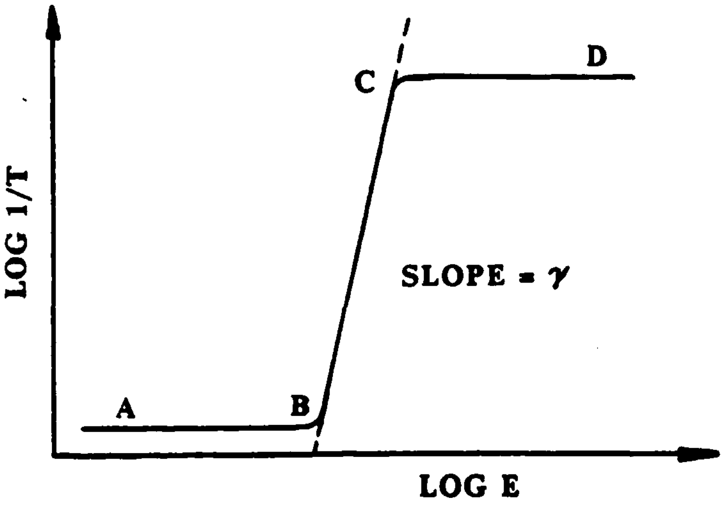


Fig. 1 Desired Input-Output Characteristic of a Bistable Optical Circuit.

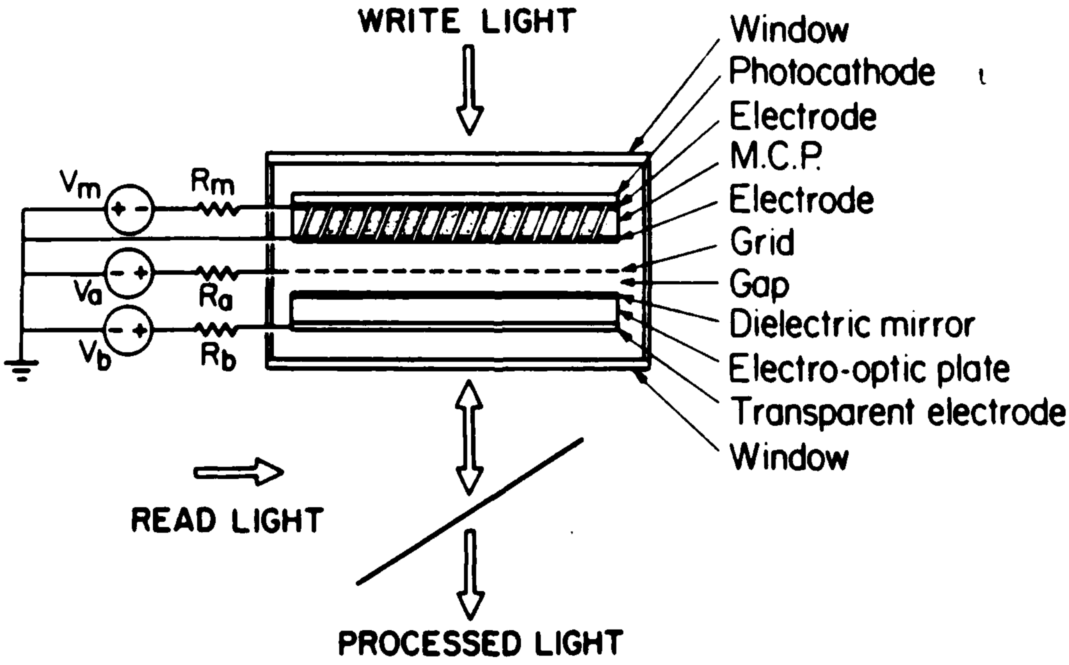
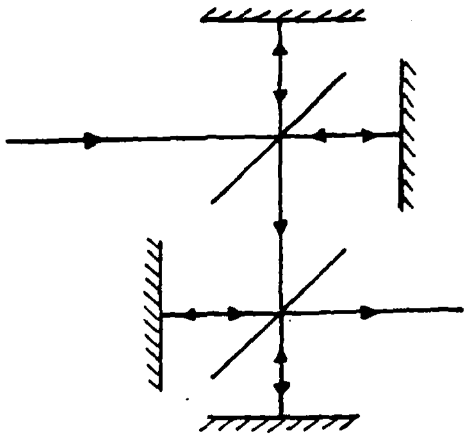
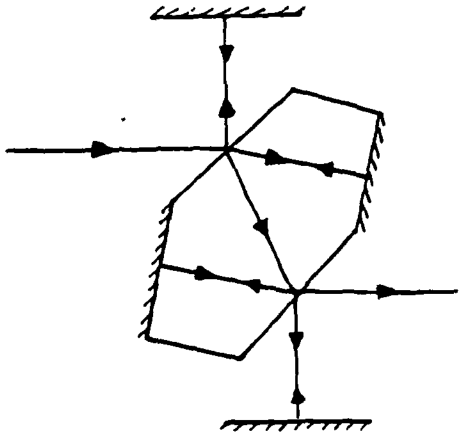


Fig. 2. The Microchannel Spatial Light Modulator.



3(a)



3(b)

Fig. 3(a) The laser beamlet array generator (BAG), based on two cascaded Michelson interferometers.

Fig. 3(b) Hexaprism version of the laser beamlet array generator.

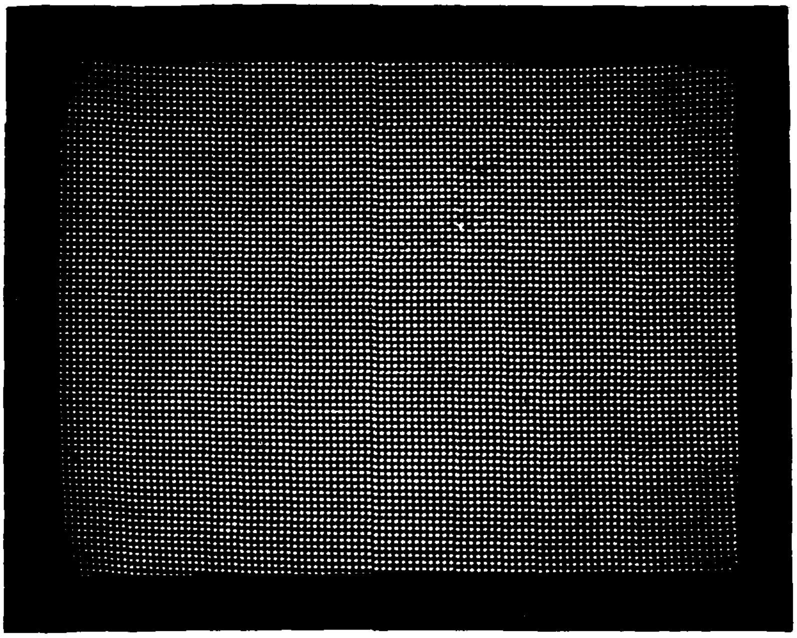


Fig. 4(a) Photograph of an array of beamlets from a BAG.

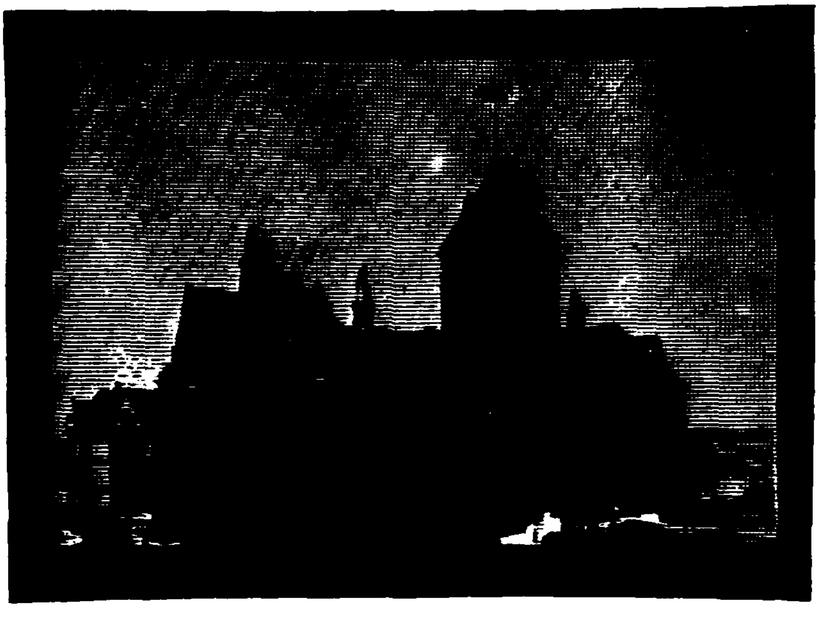


Fig. 4(b) An image after being sampled with the array of beamlets from a BAG.

Fig. 4(b).

OPTICAL CIRCUITS

The laser beamlet array generator and the MSLM can be combined in a variety of ways to realize several classes of spatially-discrete arrays of versatile, open- and closed-loop optical circuits. The uniqueness and versatility of these optical circuits arise from exploiting not only the conventional operations of image addition and multiplication of the MSLM, but also its image subtraction and nonlinear operations such as intensity thresholding and hard clipping.

General Purpose Circuits

The general purpose optical curcuit shown in Fig. 5 employs two cascaded MSLMs in which a portion of the readout signal from each device can form the input to the other. MSLM 1 behaves like a central processing unit, while MSLM 2 can be used either for storage and/or additional processing. The mirrors M_1 and M_2 are used to direct flood light beams onto the MSLMs when erasure is needed, or when the MSLMs are operating in the internal space domain processing mode.[2]

To operate the circuit shown in Fig. 5, the signal to be processed is passed through shutter S_1 and imaged onto the photocathode of MSLM 1. By appropriately gating the shutter S_2 and controlling the device voltages, MSLM 1. will perform linear and nonlinear operations such as image addition, subtraction, thresholding, hard clipping, and contrast reversal on the input signal. Part of the readout light from MSLM 1 is tapped off (at bs_3) to be the output, and part can be stored or further processed by MSLM 2 by opening S_4 and appropriately operating the shutter S_5 while controlling the device voltages. Part of the processed or stored information from MSLM 2 can be extracted as a second output of the system through bs_7, and part of this signal can be fed back to form a new input to MSLM 1 and/or to readout MSLM 1 by opening shutter S_6 and/or S_7 respectively. The optical isolator I_1 prevents the processed output light from MLSM 1 from feeding back to read out MSLM 2. When image sampling is desired or if processing spatially-discrete arrays of signals the beamlet array generator is inserted at the location shown.

With S_6 and S_7 closed, the module is an open loop system and will perform all the conventional open loop operations such as image addition and subtraction, contrast reversal, as well as nonlinear operations such as intensity thresholding, hard clipping, binarization, halftoning, and binary level logic operations such as OR, AND, NOR, NAND AND XOR.[1]

For real-time halftoning, for example, the open-loop configuration is used in conjunction with the laser beamlet array generator. The image to be processed is injected into MSLM 1, and is read out with the collimated beamlet array and stored in MSLM 2. After intensity thresholding is performed in MSLM 2 on the sampled image, the readout beam from MSLM 2 will be a halftone image of the original. Note this operation is accomplished without the use of the conventional halftone contact screens.

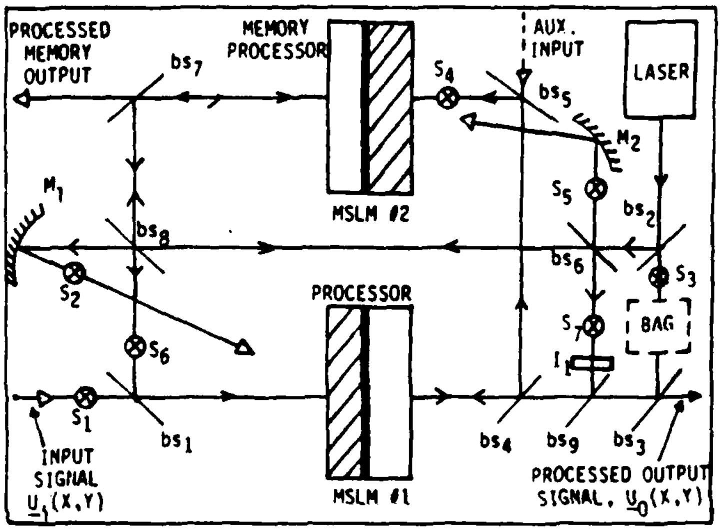


Fig. 5. A general purpose versatile optical circuit.

With S_6 closed and S_7 open, the module is a closed loop system and, for example, it will perform such non-linear operations as image intensity squaring, cubing (in general $|\underline{U}_i(x,y)|^{2n}$) where $\underline{U}_i(x,y)$ is the complex envelope of the signal to be processed and n is the integer. Hence, with a large dynamic range time-integrating detector array (e.g., CCD) in the path of the processed output light, this system can in principle perform such functions as: log[1 + I(x,y)], exp[I(x,y)] and cos[I(x,y)] subject to the dynamic range limitations of the modulator. Alternatively, with S_7 closed and S_6 open, the module can be operated in a 2-D bistable mode.

BISTABLE CONFIGURATIONS

Illustrated in Figs. 6 and 7 are a second class of all-optical circuits which are capable of performing real-time wavefront phase measurement, compensation and conjugation, as well as functioning as 2-D phase and amplitide-driven bistable arrays, respectively.

Phase-Driven Bistability

In phase-driven bistability, one seeks a transfer characteristic of the type shown in Fig. 1 where the readout light of the system switches state when the phase of the input object exceeds some threshold. To achieve 2-D arrays of phase-driven bistable circuits with an MSLM, an IPL system is employed in combination with the beamlet array generator and a variable uniform phase retardation plate is placed in the plane P_2, as shown in Fig. 7. Thus the shutter S in Fig. 6 must be opened to provide an array of reference beamlets. As shown in earlier work,[4-5] the feedback signal intensity on the photocathode of the modulator for this configuration will be dependent only on the phase of the input signal beam. For real-time spatially-continuous phase measurement, compensation and conjugation, the beamlet array generator can be omitted and the shutter S is opened so that the system becomes a standard interference phase loop (IPL).[4-5]

Amplitude-Driven Bistability

The optical feedback configuration of Fig. 6 has been suggested as a means for (1) compensating phase variations introduced by the modulator crystal, and (2) realizing bistable operation. Phase compensation is especially important for high finesse Fabry-Perot MSLMs which would otherwise have extremely stringent crystal plane-parallelism requirements for image processing, unattainable by present polishing technology.

In amplitude-driven bistability, one seeks a transfer characteristic of the type shown in Fig. 1 in which the readout light of the system switches state when the object intensisity transmittance exceeds a certain threshold. The system or image under investigation is placed either in the input plane (at P_1) or in the feedback path at (P_2). A variable spatially-uniform amplitude attenuation plate is also placed in the feedback path at P_2 to control the feedback level. The bistable output in recorded at the plane A. Note that in this case the intensity of the signal fed back is proportional to the intensity of the output signal beam. For arrays of bistable elements a beamlet array generator is used. Because of the extremely high exposure

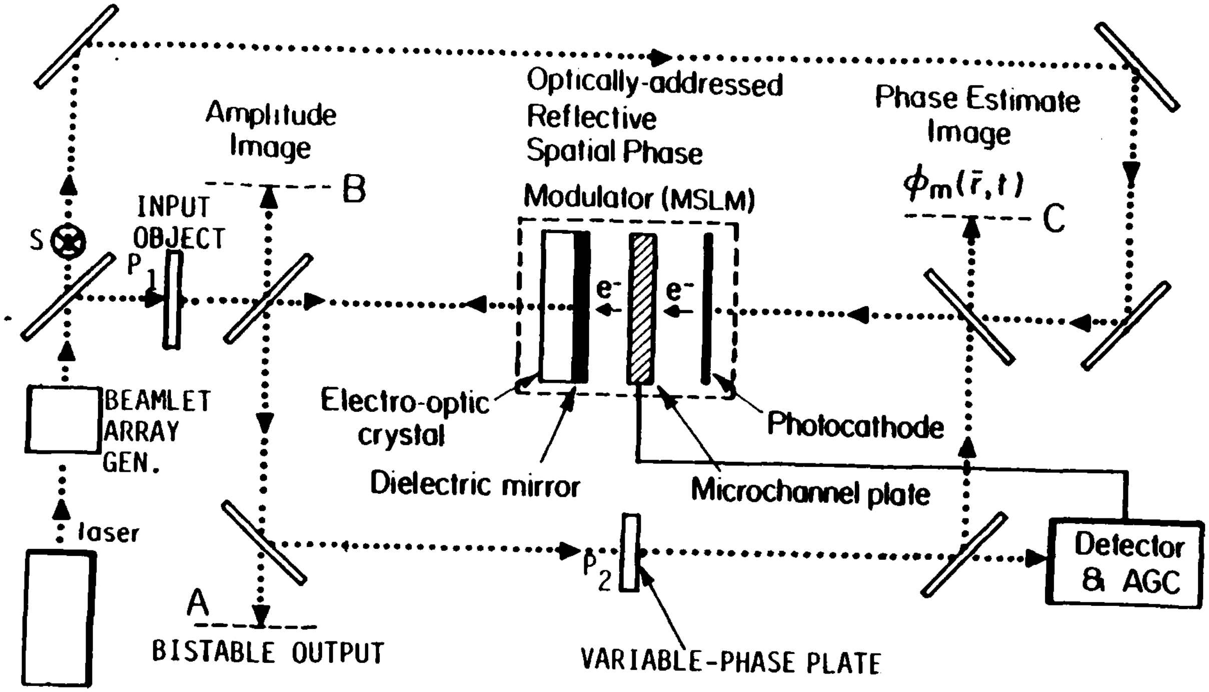


Fig. 6. IPL optical circuit modified for phase-driven bistable array operation of the MSLM.

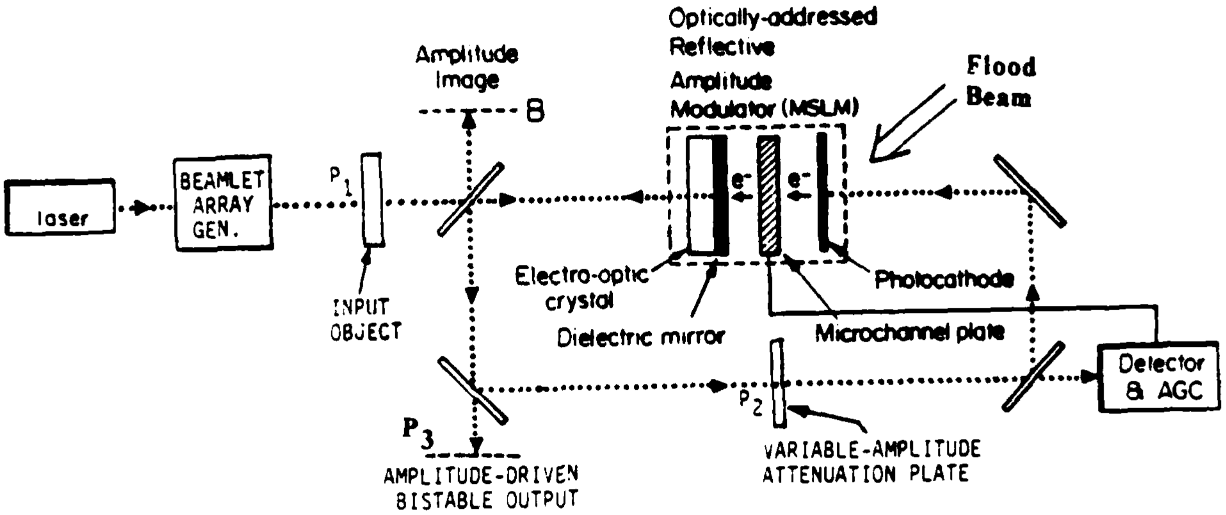


Fig. 7. **Feedback set up for amplitude-driven bistable array operation of the MSLM.**

sensitivity of the MSLM, the calculated switching energies for these amplitude-driven bistable circuits are on the order of 10^{-14} J/beamlet with current MSLMs.

In Fig. 7 crossed polarizers are omitted for z-cut crystals since these are read out interferometrically. The MSLM may be operated in the closed loop (feedback) mode by opening the shutter S, and in the open loop mode by closing the shutter. The MSLM output is monitored in the plane P_3 and is equal to some fraction of the feedback signal. A procedure for operating both a standard or Fabry-Perot MSLM in this configuration would be: 1) compensate, closed loop, 2) bias crystal, open loop, 3) write nonlinear image, closed loop.

Phase compensation and bistable operation are accomplished as follows: (1) Theinput object is removed, the feedback path is closed (s open) and the device is operated, preferably in the electron depletion mode, until the readout intensity in plane P_3 is uniformly zero.

In this closed loop operation, areas of non-zero transmittance on the crystal will result in feedback to the photocathode and deposition of surface charge on the corresponding areas of the crystal so as to drive the transmittance of the entire modulator at that region to zero. The intrinsic phase variations of the MSLM are thus compensated with the appropriate amount of surface charge and a stable equilibrium state is attained.

The device should now be biased to a bright background by operation in the open loop mode (S closed) and again flooding the photocathode with light and adjusting V_b to add or remove a uniform charge density while still maintaining the phase compensation charge profile. The flood light is then shut off and V_b set to a large enough voltage such that the gap voltage V_g remains high enough to ensure linear operation throughout the compensation process.

The image transparency to be processed is the placed in the input plane P_1 and transferred by lens L_o onto the crystal with the shutter S open. The resulting feedback signal is once again reimaged via the feedback path onto the photocathode. In this case, for a given fixed incident intensity on the input image, the high transmittance regions of the input image will at first result in strong feedback and rapid deposition of sufficient surface charge on the crystal so as to drive the modulator transmittance of those points to zero (stable equilibrium) within the allowed feedback time t_w. On the other hand, low transmittance regions of the input image result in weak feedback, which will in turn result in very small amounts of additional charge deposition on the crystal. Thus, at the end of the feedback time t_w the transmittance of the modulator at the low transmittance regions of the input image remains high. The MSLM stores these charge patterns and when read out in the open loop mode using a uniform source, a bistable image of the input transparency results.

Theory of Amplitude-Driven Bistability

The theory of the MSLM operation in this optical feedback configuration can best be modelled by considering the behavior of a single pixel of the modulator and neglecting any transverse effects from neighbouring pixels.

The intensity of the feedback signal, $I_f(t)$ can be written as,

$$I_f(t) = I_i T(t) \tag{1}$$

where I_i is the incident input intensity transferred to the crystal, T(t) is the particular time-dependent transmittance of the crystal and t is the elapsed feedback time. $I_0(t)$, the output signal is some fraction of the feedback signal (see Fig 8).

For a standard MSLM with an obique cut $LiNbo_3$ crystal the transmittance T(t) is given by,

$$T(t) = \sin^2 \Gamma(t)/2 = \sin^2 [\pi\, \sigma(t)(n^3{}_e r'_y - n_0{}^3 r'_x)/2\lambda C] \tag{2}$$

where $\sigma(t)$ is the total accumulated surface charge density on the crystal at time t and C is the capacitance per unit area of the crystal. In the linear mode of operation the incremental charge accumulated in the time dt is given by

$$d\sigma(t) = (\eta eG/h\nu)I_f(t)dt \tag{3}$$

where η is the quantum efficiency of the photocathode, G is the gain of the microchannel plate, h is Planck's constant, ν is the frequency of the light and e is the charge of an electron. The total charge density integrated on the surface of the crystal at time t is therefore,

$$\sigma(t) = \int_0^t (\eta eG/h\nu) I_f(t')dt' + \sigma_0 \tag{4}$$

where σ_0 is the initial charge on the crystal. Eqn. 1 can now be rewritten as,

$$I_f(t) = I_i \sin^2 [B\int_0^t I_f(t')dt' + \Phi_0/2] \tag{5}$$

where Φ_0 is the phase retardation due to σ_0 and B is a lumped MSLM constant. For the purposes of this analysis, the lumped constant B, which acts simply as a scaling parameter, is assumed to be equal to 1. The initial conditions at t = 0 is,

$$I_f(t{=}0) = I_i \sin^2(\Phi_0/2) \tag{6}$$

which is the reflection due to the initial bias of the crystal. Notice that at time t_f,

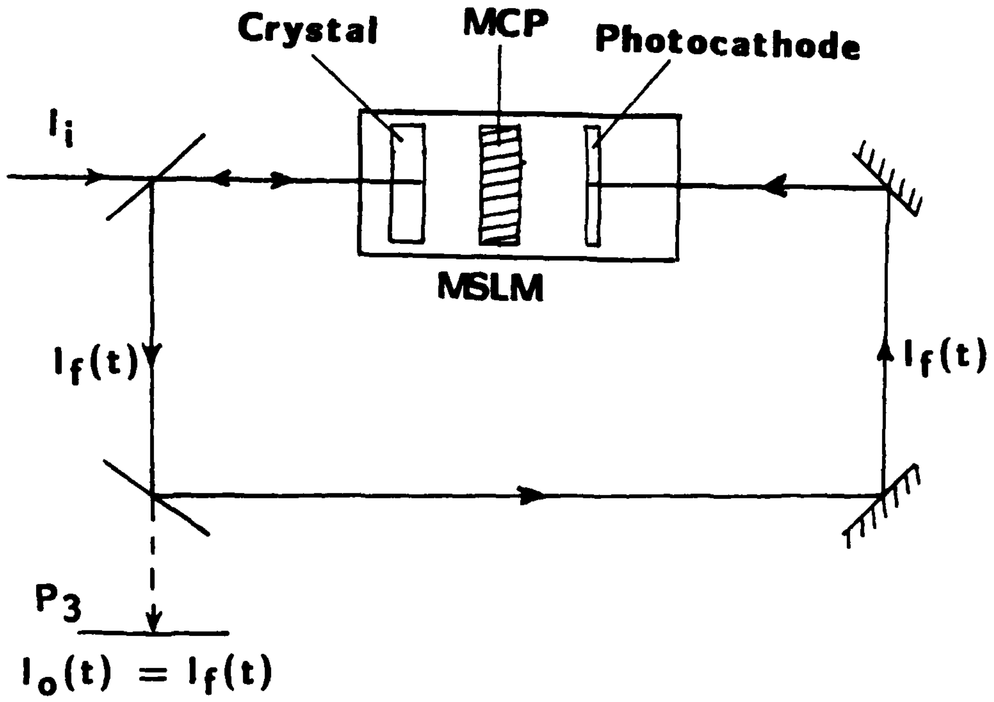


Fig. 8. Functional block diagram of the amplitude-driven bistable configuration.

$$I_f(t_f) = I_i \sin^2 [\int_0^{t_f} I_f(t')dt' + \Phi_0/2] = 0 \qquad (7)$$

if

$$\int_0^{t_f} I_f(t')dt' + \Phi_0/2 = p\pi \text{ ; where } p = 0, \pm 1, \pm 2, \ldots \qquad (8)$$

and moreover,

$$0 \leq \int_0^{t_f} I_f(t')dt' = p\pi - \Phi_0/2 \leq \pi \qquad (9)$$

Once the feedback signal $I_f(t)$ is driven to zero at time t_f, no further charge deposition occurs, so that,

$$I_f(t) = 0 \text{ for all } t \geq t_f \qquad (10)$$

and stable equilibrium is achieved. Notice that for phase compensation, the shutter S must remain open long enough to allow the entire crystal to achieve equilibrium. This behavior is monitored in the output plane P_3.

Alternatively, for nonlinear processing (e.g. bistable operation) feedback is allowed only long enough for areas of the input with transmittance above some threshold to be driven to equilibrium. The threshold transmittance is determined by the feedback time t_w. In the nonlinear processing case, t_w is such that,

$$\int_0^{t_f} I_f(t')dt' + \Phi_0/2 \cong \begin{cases} p\pi, & \text{for high input transmittance} \\ \Phi_0/2, & \text{for low input transmittance} \end{cases} \qquad (11)$$

Thus the modulator transmittance will be given by,

$$T(t_w) \cong \begin{cases} 0, & \text{for high input transmittance} \\ \sin^2(\Phi_0/2), & \text{for low input transmittance} \end{cases} \qquad (12)$$

and the desired bistable image will be attained upon readout in the open loop configuration with a uniform source.

The dynamic behavior of the feedback signal is best analysed by numerically calculating and plotting $I_f(t)$. From the fundamental theorem of calculus, Eqn. 5 can be estimated as follows,

$$I_f\ (t = N\Delta t) = I_i \sin^2 [\sum_{n=1}^{N} I_f((n-1)\ \Delta t)\Delta t + \Phi_0\ 2] \qquad (13)$$

where the feedback signal is considered constant during the incremental time interval $\Delta t = t/N$; N is the number of time intervals. The initial condition is still given by Eqn. 6.

Equation 13 can be analysed for 2 cases: 1) varying the initial bias phase Φ_0 due to the crystal nonuniformities, and (2) varying the input intensity I_i of transmitted by the input transparency.

Figure 9(a) contains plots of $I_f(t)$ for the same input intensity I_i, with different initial phases Φ_0 as a parameter. This is representative of the dynamic behaviour of different areas of a nonplane-parallel crystal, each with some intrinsic initial phase, during phase compensation. Notice that since the charge added is proportional to the instantaneous intensity of the feedback signal, the modulator output/transmission as a function of time will rise faster than the characteristic sine-squared curve until its argument approaches the next highest integer multiple of π above the initial $\Phi_0/2$ (see Eqns. 8,9). The transmission will then decay to zero faster than the characteristic sine-squared curve. Notice that in cases where $\Phi_0/2$ occurs before a maximum of modulator transmission, the feedback signal will actually increase first before decaying to zero.

The effect of different input intensities for the same initial phase Φ_0 on $I_f(t)$ is shown in Fig 9(b). and the effect on the modulator transmittance T(t) is shown in Fig 9(c). In this case, Φ_0 is set at π radians and the relative values of the input intensities varied. Notice that $I_f(t)$ (and thus, T(t)) decays to zero more rapidly for higher input intensity. This supports the discussion on the use of this feedback configuration for nonlinear applications. At some time t_w, indicated in Fig. 9(c) regions with high I_i are driven to zero transmittance, while those with low I_i leading to very little additional charge deposition, maintain relatively high modulator transmittance.

A similar analysis has been done for the dynamic behaviour of the Fabry-Perot MSLM. For this version of the device, the transmittance T(t) is given by

$$T(t) = \frac{4R\sin^2(\Phi_0/2)}{(1-R)^2 + 4R\sin^2(\Phi_0/2)} \qquad (14)$$

where Φ is the total absolute phase due to the crystal, and R is the reflectivity of the crystal. Thus, Eqn. 1 can be rewritten as

$$I_f(t) = I_i \frac{4R\sin^2[B'\int_0^t I_f(t')dt' + \Phi_0/2]}{(1-R)^2 + 4R\sin^2[B'\int_0^t I_f(t')dt' + \Phi_0/2]} \qquad (15)$$

where Φ_0 is again the initial bias phase and B' is a lumped device constant Once again, B' is a scaling parameter and is assumed equal to 1. The initial condition is,

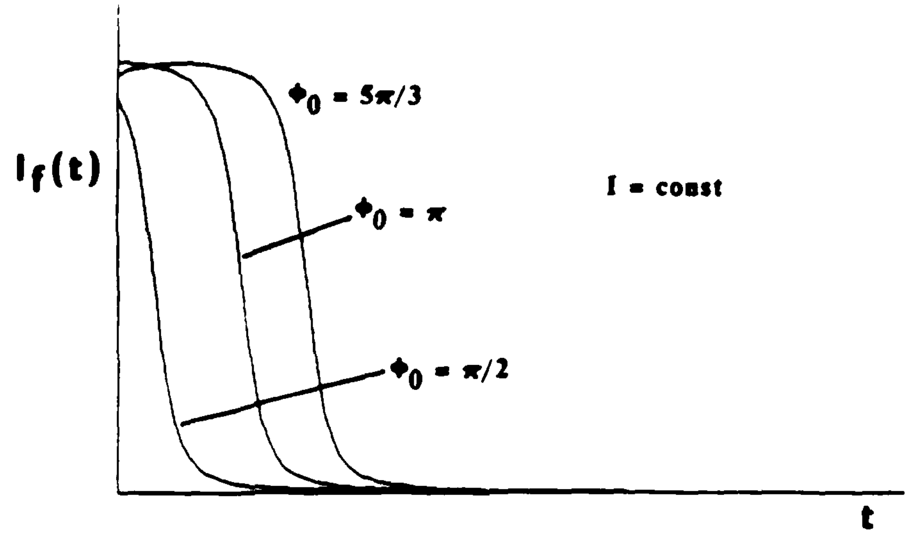


Fig. 10(a) Bistable output intensity $I_f(t)$ of a Fabry-Perot MSLM for constant input intensity I_i with initial phase Φ_0 as a parameter.

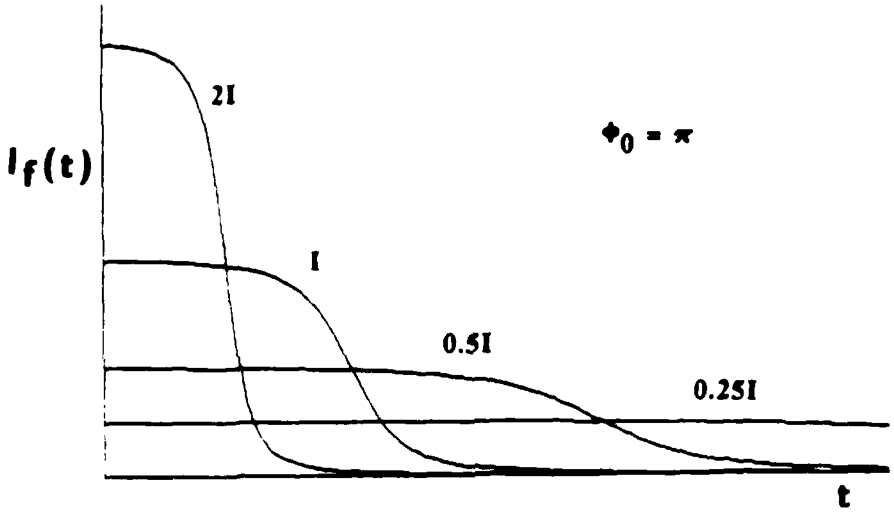


Fig. 10(b) Bistable output intensity $I_f(t)$ of a Fabry-Perot MSLM for constant initial phase $\Phi_0 = \pi$ and with input intensity I_i as a parameter.

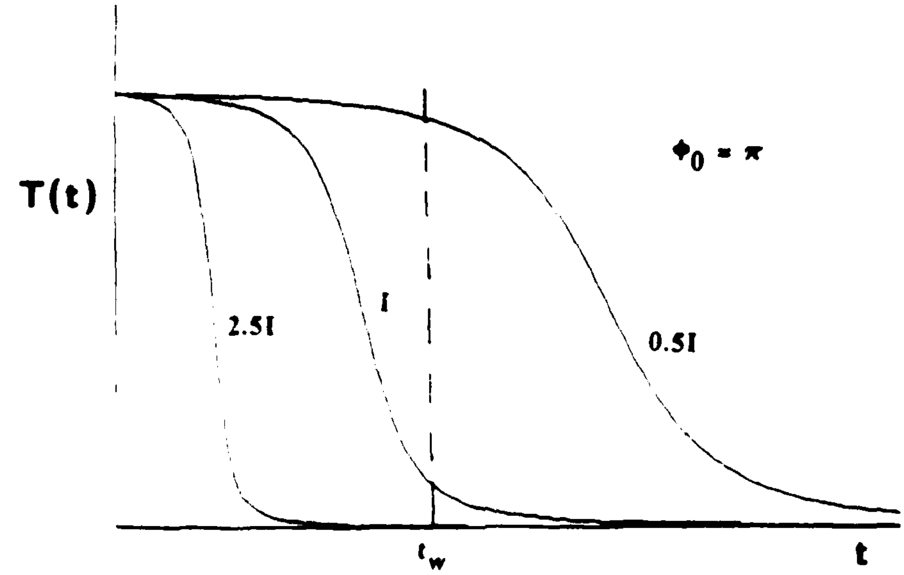


Fig. 10(c) Corresponding modulator transmission T(t) for a Fabry-Perot MSLM.

$$I_f(o) = \frac{I_i 4R \sin^2(\Phi_0/2)}{(1-R)^2 + 4R\sin^2(\Phi_0/2)} \qquad (16)$$

The feedback signal will again equilibrate at zero when $t \geq t_f$, where t_f is still as defined in Eqn. 8 . The Fabry-Perot device will operate in much the same manner as the standard device.

The high finesse Fabry-Perot device will compensate for crystal deviations from plane-paralleism if sufficient feedback time is allowed, and can achieve nonlinear operation when the feedback time is appropriately limited. Equation 15 was analysed numerically and plots were made for: (1) $I_f(t)$ with constant I_i and different Φ_0 (shown in Fig. 10(a)), and (2) $I_f(t)$ and T(t) constant Φ_0, different relative I_i (shown in Figs. 10(b) and 10(c)).

Notice that the intrinsic nonlinear characteristics of a high finesse Fabry-Perot crystal (R = 0.7) result in similar nonlinear behavior of the feedback signal and the modulator transmittance. In nonlinear operation, the threshold input transmittance can be precisely determined by t_w, and the resulting halfone image is significantly improved over that obtained with a standard device.

The disadvantage of using this type of feedback configuration to compensate for crystal deviations from plane-parallelism by charge deposition, is that there is a trade-off in dynamic operating range. Experimentally the safe dynamic operating range of current MSLM devices is about 4kV; this limitation is imposed by arcing between components and stresses due to high electrostatic forces. Consider a standard device with $V\pi R$ = 1250V; if the MSLM to be compensated has a crystal with a plane-parallelism factor, a = $\lambda/4$, then the crystal will require about $V_{\pi R}/2$ = 625V equivalent charge compensation in some areas. Thus about 15% of the available operation range is utilised in compensation, leaving over 3kV with which to work. This is not unreasonable for a crystal with $V_{\pi R}$ = 1250 (full modulation can be realized for a grid voltage of 1-2 kV).

However, the compensation trade-off is much more severe in the case of the Fabry-Perot. Consider a device with a z-cut lithium niobate crystal, $V\pi$=3100V; once again assuming a = $\lambda/4$, the crystal will require $V_{\pi R}/2$ = 1550V equivalent charge compensation. This is about 40% of the available operating range, leaving only about 2450V with which to work. Although this range is greater than the voltage required for a light to dark transition (600-800V; R = 0.7), if the grid voltage is 1-2 kV, no allowance would have been made for the fact that the precise thickness of the crystal could be as much as (n + 1/2)λ, where n is an integer and λ is the wavelength of the readout light. Thus, the crystal would be biased in the neighborhood of the central flat region of the Fabry-Perot characteristic. This additional fraction of a wavelength must be accounted for experimentally by properly biasing the operating voltages and in the worst case can require as much as V_π = 3100V.

The ability to compensate internal device phase variations is critical for image processing with a Fabry-Perot MSLM using currently available crystals. However, the amount of phase compensation that can be tolerated while leaving sufficient operating range, is limited by the precise deviation of the crystal thickness from an integer multiple of λ, and the initial voltage limitations imposed on the device.

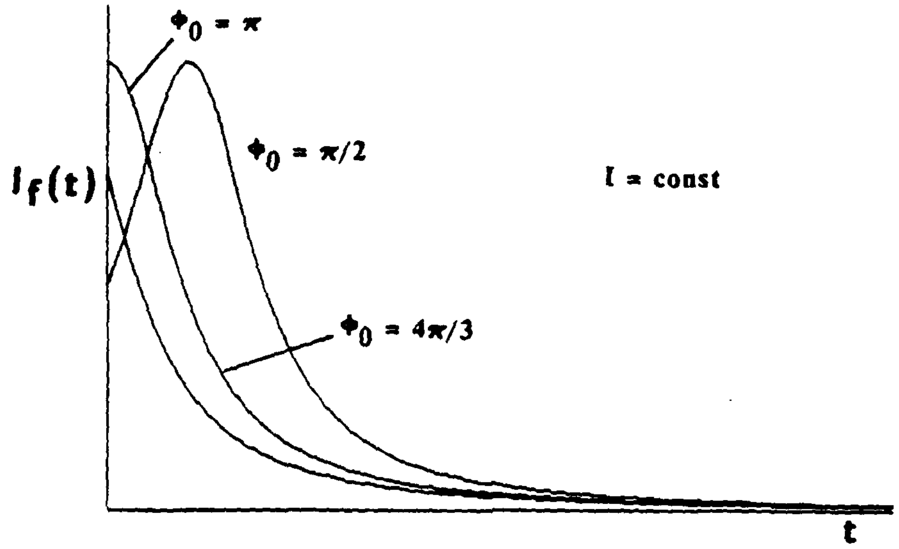


Fig. 9(a) Bistable output intensity $I_f(t)$ of a standard MSLM for constant input intensity I_i with initial phase Φ_0 as a parameter.

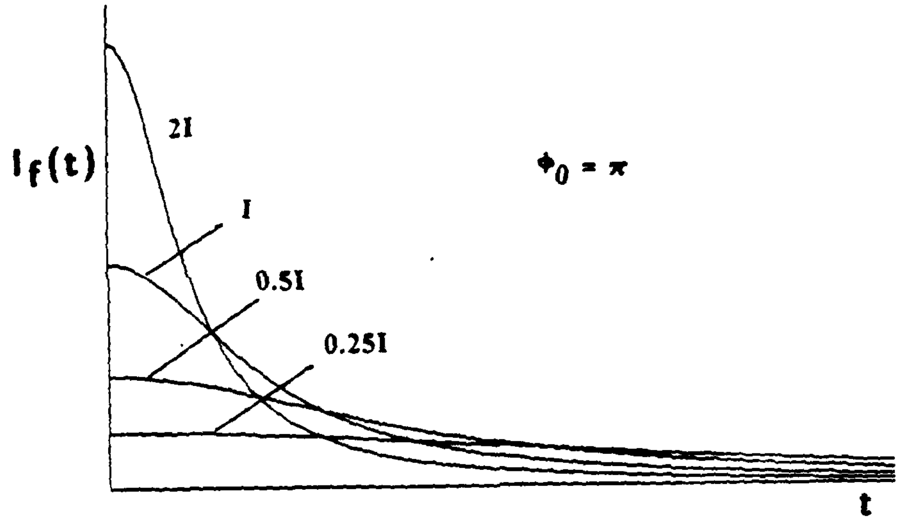


Fig. 9(b) Bistable output intensity $I_f(t)$ of a standard MSLM for constant initial phase $\Phi_0 = \pi$ and with input intensity I_i as a parameter.

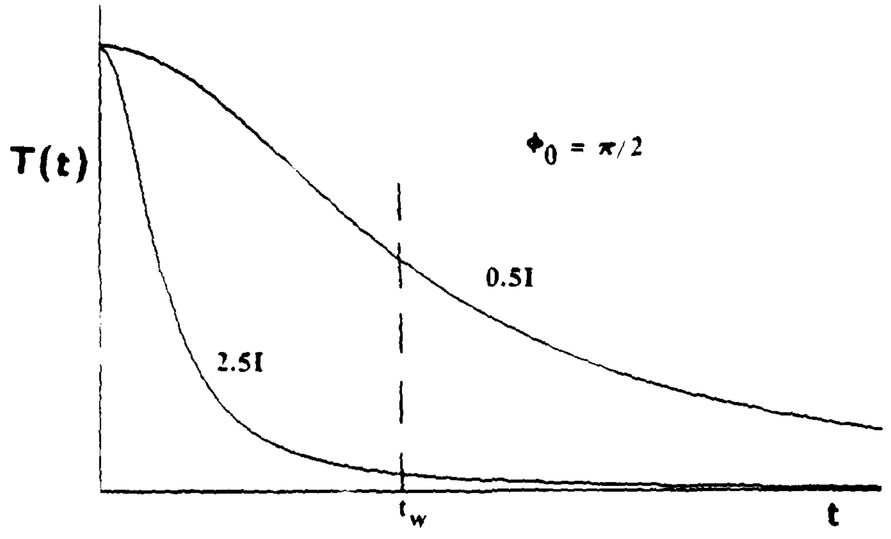


Fig. 9(c) Corresponding modulator transmission T(t) for a standard MSLM.

SECTION II - APPLICATIONS OF THE BTS IN OPTICAL PROCESSING

INTRODUCTION

To overcome the accuracy and dynamic range limitations of analog optical systems, digital optical systems have been proposed for a wide variety of applications in numerical optical processing. These applications include such areas as real and complex number arithmetic and matrix multiplication. In nearly all of these applications, the conventional binary system has been the digital system by choice.

In this paper, the balanced ternary system (BTS) is considered as an alternative to the binary system for use in simple optical processors. The properties of the BTS are first reviewed and a simple binary matrix formulation for BTS number is developed for representing bipolar and complex integers. We then describe an optical implementation which shows that the BTS provides digital accuracy and larger dynamic range while using less space-bandwidth when multiplying integers. Finally, we propose an example of an optical architecture which uses the BTS for vector-matrix multiplication.

BALANCED TERNARY SYSTEM

The balanced ternary system (BTS) is described in detail by Knuth [1]. It is based on radix 3 and consists of the digits $\bar{1}$, 0, and 1 (called trits) where $\bar{1}$ = -1. Given P ternary digits, 3^P real integers can be represented ranging from $-(3^P-1)/2$ to $+(3^P-1)/2$. To convert a balanced ternary number into decimal, the standard method is employed whereby the M^{th} digit is weighted by 3^{M-1}. Thus, the number

$$\begin{aligned} 10\bar{1}1_{BTS} &= (1)3^3 + (0)3^2 + (-1)3^1 + (1)3^0 \\ &= 27 - 3 + 1 \\ &= 25. \end{aligned}$$

More examples of balanced ternary numbers are shown in Table 1.

The basic operations of addition, subtraction, and multiplication in the BTS are nearly identical to that of the binary system. The exceptions arise primarily in the carries when adding two numbers. Table 2 lists the addition rules for the BTS. The multiplication of two numbers will in general involve multiplying by +1, 0, and -1, and then adding the partial results. For example, the product (5)(-6) = $(1\bar{1}\bar{1})(\bar{1}10)_{BTS}$ is calculated by

$$\begin{array}{r} 1\bar{1}\bar{1} \\ \times\ \ \bar{1}10 \\ \hline 000 \\ 1\bar{1}\bar{1} \\ +\ \bar{1}11 \\ \hline 0\bar{1}0\bar{1}0 \end{array}$$

which equals -30.

Table 1

Examples of Balanced Ternary Numbers

Decimal	Binary	BTS
2	00000010	000$\bar{1}$1
-1	11111111	0000$\bar{1}$
35	00100011	0110$\bar{1}$
-35	11011101	0$\bar{1}\bar{1}$01

Table 2

Rules for the Addition of in
the Balanced Ternary System

Addition Operands	Result w/Carry
$\bar{1} + \bar{1}$	$\bar{1}1$
$\bar{1} + 0$	$0\bar{1}$
$\bar{1} + 1$	00
0 + 0	00
0 + 1	01
1 + 1	11

DIGITAL OPTICAL IMPLEMENTATION OF THE BTS

Implementing any ternary system electronically would be difficult since it would require a tristable device. However, an elegant optical digital implementation exists if the data field is extended to two dimensions. In general, a real integer can be considered as a vector of digits. This vector can be decomposed into the product of a base digit vector containing all possible digits and a binary matrix. For the BTS, the base digit vector is [1 0 $\bar{1}$] which can be reduced to [1 $\bar{1}$] since multiplication by zero is zero.

With the base digit vector fixed, the binary matrix becomes the variable entity. For example, the number -313 is represented as

$$-313 = (\bar{1}\bar{1}011\bar{1})_{BTS}$$

$$\rightarrow [\bar{1} \;\; \bar{1} \;\; 0 \;\; 1 \;\; 1 \;\; \bar{1}]$$

$$\rightarrow [1 \;\; \bar{1}] \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The first row in the binary matrix corresponds to the positive values of the radix multipliers. In the same manner, the second row maps to the negative values. In general, the radix multipliers are

$$\begin{bmatrix} . & . & . & 243 & 81 & 27 & 9 & 3 & 1 \\ . & . & . & -243 & -81 & -27 & -9 & -3 & -1 \end{bmatrix}$$

A number is formed by simply adding up the appropriate multipliers.

In the optical implementation of a number in the BTS, a light beam having constant intensity would be positioned, in a relative sense, at each location of a required multiplier. Thus, each "1" in the binary matrix representation becomes a beam of light in the optical implementation. To read the value of a number, an array detector having the dimensions of the binary matrix would be used and the output would be processed electronically. At the input to a system, there would be no redundancy whereby complement multipliers, say 27 and -27, are selected. In this digital optical implementation, it would be necessary to consider only the binary matrix representation of the number; that is, the base digit vector $[1 \; \bar{1}]$ can be implied.

The multiplication of two balanced ternary numbers stored in binary matrix form is the two-dimensional spatial convolution of the matrices. For the output matrix to have the same multiplier orientation as the input binary matrices, the spatial origin of the input matrices should be placed at the +1 multiplier. Then, the origin of the output matrix will also be located at the +1 multiplier. Let A and B represent two integers and let their BTS representations using binary matrices be denoted by $\underline{A}$ and $\underline{B}$, respectively. The product AB = C has a BTS binary matrix given by $\underline{C} = \underline{A} * \underline{B}$ where * represents the two-dimensional spatial convolution operation. If $\underline{A}$ is a 2 x N matrix and $\underline{B}$ is a 2 x M matrix, the output of $\underline{C} = \underline{A} * \underline{B}$ will be a 3 x (N+M-1) mixed binary matrix. The first row of the output is a 1 row and results from the multiplication of the 1 rows of $\underline{A}$ and $\underline{B}$. The second row is a $\bar{1}$ row and is the sum of two components. One of these components is the product of the 1 row of $\underline{A}$ and the $\bar{1}$ row of $\underline{B}$; the other component is due to the multiplication of the $\bar{1}$ row of $\underline{A}$ with the 1 row of $\underline{B}$. The new third row represents the multiplication of the $\bar{1}$ rows of $\underline{A}$ and $\underline{B}$, producing a 1 row. This third row can be combined with the first row, if desired. If, however, the third row is kept separate, the value of $\underline{C}$ (in the BTS) can be obtained by extending the base digit vector such that

$$\underline{C} = [1 \;\; \bar{1} \;\; 1] \begin{bmatrix} 3 \times (N+M-1) \\ \text{Mixed Binary} \\ \text{Output Matrix} \end{bmatrix}$$

For example, if A = -15 and B = 5, then

$$\underline{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \underline{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

The convolution of $\underline{A}$ with $\underline{B}$ gives $\underline{C}$ where

$$\underline{C} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

In the optical implementation, $\underline{C}$ is composed of light beams with intensity levels nI_0 at each location for the n^{th} digit. That is, locations with a "1" correspond to a light beam intensity of I_0 while those with a "2" have intensity $2I_0$. To convert $\underline{C}$ to the number C, the beams of light are directed onto an array detector whereby the intensities nI_0 are transformed into voltages levels nV_0 from a reference voltage V_0. The actual value of C can then be electronically processed according to the following mathematical steps:

$$\begin{aligned} C &= 1 \;\; \bar{1} \;\; 1 \;\; \underline{C} \\ &= \bar{1} \;\; 2 \;\; 1 \;\; \bar{2} \;\; \bar{1} \;\; 0 \\ &= \bar{1}21\bar{2}\bar{1}0 \;\; \text{BTS} \\ &= \bar{1}01\bar{1}0 \;\; \text{BTS} \\ &= -75 \end{aligned}$$

where the second and third lines are in mixed ternary form with $\bar{2} = -2$. Note that when "1" bits appear in both the 1 and $\bar{1}$ rows of a column in the binary matrix, they cancel each other and can be replaced with zeros.

The conversion from mixed ternary digits to the standard BTS digits can be derived from the addition rules presented in Table 2. As with other base systems, the procedure is executed from right to left. If the digit is a "2", "3", or "4", subtract 3 from it and carry $\bar{1}$ to the next column to the left. If the digit is "6", "7", or "8", the process is doubled so 6 is subtracted from the digit and $\bar{2}$ is carried to the next column. In general, for a positive digit P in the interval k+1 ≤ P ≤ k+3, the value -3k is added to the digit and k is carried over to the next column. From the symmetry of the BTS system, the algorithm for negative digits is the same as for positive digits.

except the signs of the values added and carried are reversed. Thus, for a negative digit Q in the interval $-(k+3) \leq Q \leq -(k+1)$, the value $+3k$ is added to the digit and the carry is k.

Complex Number Representation

To represent complex numbers in this implementation, the base digit vector and binary matrix must be modified to incorporate the real and imaginary components. The logical extension is to expand the base digit vector to $[1\ \bar{1}\ j\ \bar{j}]$ and the binary matrix to a 4 x N array. The first and second rows of this matrix correspond to the real part while the third and forth rows encode the imaginary part. The radix multipliers for each bit in the binary matrix now become

$$\begin{bmatrix} . & . & . & 243 & 81 & 27 & 9 & 3 & 1 \\ . & . & . & -243 & -81 & -27 & -9 & -3 & -1 \\ . & . & . & j243 & j81 & j27 & j9 & j3 & j \\ . & . & . & -j243 & -j81 & -j27 & -j9 & -j3 & -j \end{bmatrix}$$

For example, the complex number $25 - j32 = (10\bar{1}1 + j11\bar{1}\bar{1})_{BTS}$ has the binary matrix expansion

$$25 - j32 = (10\bar{1}1 + j11\bar{1}\bar{1})_{BTS}$$

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

The multiplication of complex numbers in the BTS format therefore follows the same convolution principles as for real integers. However, the process is more tedious because of the need to keep track of the real and imaginary products and to combine them appropriately.

BALANCED TERNARY SYSTEM VS. BINARY SYSTEM

On a mathematical basis, the BTS has several interesting properties which the binary system lacks. With the BTS, positive and negative numbers can be represented equally well. The sign of a number is the same as the sign of the leftmost digit, eliminating the need for a sign bit. The operation of rounding is identical to replacing the undesired digits with zeros. The complement of a number is formed by simply interchanging $\bar{1}$ and 1. On average, balanced ternary numbers use a factor of (ln 3)/(ln 2) less digits than the same numbers encoded in binary [1].

In terms of their optical digital implementations, the BTS offers several advantages over the binary system (with negative integers encoded using two's complement notation). In the following discussion, a comparison of the implementation formats for the two systems is first presented. Then, the method for multiplying two integers with complete parallelism is discussed. In this comparison, it is shown that the BTS provides larger dynamic range for the amount of space-bandwidth utilized than the binary system.

As with the mathematical formulation, the complement of a complex BTS number is easily obtained by either interchanging rows 1 and 2 and rows 3 and 4 of the binary matrix or by reversing the implied base digit vector to $[\bar{1}\ \ 1\ \ \bar{j}\ \ j]$. In the optical implementation, the use of this latter option has the benefit of not requiring rows of light beams to be interchanged or switched to affect the complementing process, provided the rest of the system is designed for the modified base digit vector.

Secondly, the BTS format uses light power more efficiently than the binary system on average. Each bit in either implementation corresponds to a beam of light, resulting in "1" bits requiring power. When representing small negative numbers, the binary system will need several "1" bits to retain the sign information. This can be seen by considering, for example, the representations for +1 and -1:

Decimal	BTS	Binary System
+1	$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	00000001
-1	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	11111111

Additionally, because the binary matrix in the BTS representation is two-dimensional, it would utilize the space on a two-dimensional spatial light modulator (2-D SLM) more efficiently than the binary system. As an example, consider a 3 x 3 array of complex numbers with real and imaginary components ranging from -1000 to +1000. The BTS requires 4 x 7 bits while the binary system needs 2 x 11 bits. For the 3 x 3 array, the BTS format uses a 12 x 21 bit matrix compared to a 6 x 33 bit array for the binary system. Since 2-D SLMs currently tend to be round or square, rectangular bit arrays will fit on the available space more efficiently than linear ones. Thus, the BTS format is more space efficient than the binary system.

In the same manner, the BTS requires less space-bandwidth than the binary system since it has lowest total number of bits in a linear direction. In the previous example, the total number of bits in a linear dimension was 21 and 33 for the BTS and binary systems, respectively. In general, the resolution needed for the BTS is less than the binary system by a factor of (ln 3)/(ln 2).

When multiplying two real integers with complete parallelism, the BTS format corresponds to the true spatial convolution of their binary matrices. Strictly speaking, this is not the case with the binary system due to the presence of a sign bit. Consider the product (6)(-15). In the BTS, the convolution of the binary matrices for 6 and -15 gives

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

in mixed ternary which can be reduced to

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow -90$$

In the binary system, the convolution of the binary vectors for 6 and -15 produces

$$00000110 * 11110001 = 1222100110$$

in mixed binary which can be simplified to 10110100110_2. The final answer is obtained by keeping only the right 8 bits since the numbers were encoded using 8 bits originally. This results in the correct answer of 10100110_2 or -90. However, since a truncation was necessary in the binary case, the multiplication of two real integers does not correspond to true spatial convolution. No truncation was needed for the BTS format.

Finally, when multiplying two real integers, the BTS provides larger dynamic range while using less space-bandwidth than the binary system. The gain in dynamic range is due to the radix 3 base. The decrease in space-bandwidth is caused the lack of a sign bit and the absense of a truncation on the output of the convolution. Let A_1 and B_1 be real BTS integers, each having M digits. Their binary matrix formats are then 2 x M bit arrays. The range of integers represented by A_1 and B_1 are from $-(3^M-1)/2$ to $+(3^M-1)/2$ for a total of 3^M integers. The output $C_1 = A_1B_1$ has a binary matrix representation (in mixed binary) which is the spatial convolution of the binary matrices for A_1 and B_1 and thus has dimensions 3 x (2M-1). The corresponding output range of numbers is from $-(3^M-1)^2/4$ to $+(3^M-1)^2/4$ for a total of $(3^M-1)^2/2 + 1$ numbers.

Now, let A_2 and B_2 represent real binary integers, each having N bits. The spatial convolution to form $C_2 = A_2B_2$ produces an output containing 2N-1 bits (in mixed binary). However, this output must be truncated to N bits after the conversion from mixed binary to binary. The output range is then -2^{N-1} to $+2^{N-1}-1$ for a total of 2^N numbers. The effective input range becomes $-2^{(N-1)/2}$ to $+2^{(N-1)/2}$. Therefore, the input numbers have wasted bits which must be present to preserve the sign bit. If N is odd, there are N-[(N-1)/2+1] = (N-1)/2 wasted bits; otherwise, there are N-[N/2+1] = N/2 - 1 wasted bits if N is even. If an asymmetry in the input is allowed such that A_2 is always larger than B_2, the range for A_2 can be larger than that for B_2, provided the product A_2B_2 is within the specified output range.

Given this setup, the formats can be compared by considering the case where the widths of the input numbers are equal so M = N. Under these conditions, the output in the BTS format is a 3 x (2N-1) bit array and has a total range of $(3^N-1)^2/2 + 1$ numbers. Meanwhile, the binary format output is 2N-1 bits long which must be converted from mixed binary to binary and then truncated to N bits. The range in the binary format is thus 2^N numbers. It is convenient to define a figure of merit (FOM) relating the available dynamic range on the input compared to the space-bandwidth used. Mathematically, the FOM can be expressed as

$$\text{Figure of Merit (FOM)} = \frac{1 + (\text{Max Input Value}) - (\text{Min Input Value})}{N}$$

where N is the width of the inputs. Substituting the ranges for the two number systems, the individual FOMs can be expressed as

$$\text{FOM}_{\text{BTS}} = \frac{3^N}{N}$$

$$\text{FOM}_{\text{BINARY}} = \frac{\text{INT}(1 + 2^{(N+1)/2})}{N}$$

where INT(m) represents the largest integer less than or equal to m. Table 3 summarizes the input and output range values along with the corresponding FOMs for both formats as a function of N. In the binary case, no asymmetry is assumed so the ranges for both input numbers must be the same. Clearly, for a given input width, the BTS provides superior dynamic range while utilizing less space-bandwidth than the binary system.

Note that in this case the BTS uses three times as many bits as does the binary system on the output. However, since most 2-D SLMs have approximately equal resolutions in orthogonal planar directions and N > 3 typically, the two-dimensional format of the BTS does not produce any additional requirements on the space-bandwidth of the device being used.

If the FOM had been defined as the ratio of the input dynamic range to the total number of input bits used, the values of FOM_{BTS} would be decreased by a factor of 2. However, they would still remain far above those for the binary system. In the same manner, if the FOM had considered the output dynamic range, the BTS would again dominate because the output would not have to be truncated to the original input width as with the binary system.

Table 3

Input and Output Range Comparisons
For Equal Input Number Bit Widths

	Balanced Ternary System			Binary System		
N	Input	Output	FOM	Input	Output	FOM
2	4	16	4.5	1	(2)	1.5
3	13	169	9.0	2	(4)	1.7
4	40	1600	20.3	2	(8)	1.5
5	121	14641	48.6	4	(16)	1.8
6	364	132496	121.5	5	(32)	2.0
7	1093	1194649	312.4	8	(64)	2.4
8	3280	1.0758E07	820.1	11	(128)	2.9
9	9841	9.6845E07	2187.0	16	(256)	3.7
10	29524	8.7167E08	5904.9	22	(512)	4.6
11	88573	7.8452E09	16104.3	32	(1024)	5.9
12	265720	7.0607E10	44286.8	45	(2048)	7.6
13	797161	6.3547E11	122640.2	64	(4096)	9.9
14	2391484	5.7192E12	341640.6	90	(8192)	13.0
15	7174453	5.1473E13	956593.8	128	(16384)	17.1
16	2.1523E07	4.6327E14	2690420.0	181	(32768)	22.7

Notes:

1. The range limits above are ± figures. Thus if a range is given as 364, it is actually from -364 to +364.

2. For numbers in parentheses, the actual range is given by (P) = -P to P-1. In this case, the value +P cannot be attained by squaring the maximum input value.

3. For the Binary System, an input value can be larger than the indicated range provided the other input to the multiplication is sufficiently smaller in magnitude so that their product is within the specified output range.

SUMMARY

The balanced ternary system has been presented as an alternative to the binary system for encoding real bipolar and complex numbers. An optical implementation of this system was developed and shown to maintain digital accuracy. An extension to include bipolar complex numbers was included along with a comparison of the two number systems. It was found that the balanced ternary system offered several useful advantages over the binary system, among them being the more efficient use of power and space-bandwidth coupled with increased dynamic range when performing multiplication.

A major factor in using the balanced ternary system is the electrical conversion between it and the binary system. Aside from the limitations of current two-dimensional optical devices, the conversion performance will determine its usability.

REFERENCES

1. C. Warde, A. M. Weiss, A. D. Fisher, and J. I. Thackara, "Optical Information Processing Characteristics of the Microchannel Spatial Light Modulator," applied Optics, 20, 2066-2074, 1981

2. C. Warde, and J. I. Thackara, "Oblique-cut $LiNbO_3$ Microchannel Sp;atial Light Modulator, "Optics Letters, 7, 244-246, 1982

3. C. Warde and J. I. Thackara, "Operating Modes of the Microchannel Spatial Light Modulator," Optical Engineering, 22, 695-702, 1983

4. A. D. Fisher and C. Warde, "Simple Closed-Loop System for Real-Time Optical Phase Measurement," Optics Lett., 4, 131-133, 1979

5. A. D. Fisher and C. Warde, "Technique for Real-Time, High-Resolution Adaptive Phase Compensation, "Optics Lett., 8, 353-355, 1983.

6. D. E. Knuth, The Art of Computer Programming, Vol. 2, Addison-Wesley, Reading, MA, 1981, pp. 190-2.

SECTION III - HYBRID OPTICAL INFERENCE MACHINES

Before completion of the program, work was initiated on the design of hybrid optical engines for symbolic inference on relational data bases. The goal os to (1) increase the speed of searching and matching operations by using the parallelism of optics to perform parallel rather than sequential ssearching and pattern matching (as is customary in languages such as PROLOG) and (2) eliminate the need for backtracking through the knowledge base. The results of this investigation are summatrized in the reprint below entitled "Hybrid Optical Inference Machines: Architectural Considerations".

Hybrid optical inference machines: architectural considerations

Cardinal Warde and James Kottas

A class of optical computing systems is introduced for solving symbolic logic problems that are characterized by a set of data objects and a set of relationships describing the data objects. The data objects and relationships are arranged into sets of facts and rules to form a knowledge base. The solutions to symbolic logic problems involve inferring conclusions to queries by applying logical inference to the facts and rules. The general structure of an inference machine is discussed in terms of rule-driven and query-driven control flows. As examples of a query-driven inference machine, two hybrid optical system architectures are presented which use matched-filter and mapped-template logic, respectively.

I. Introduction

A. Definitions

Symbolic logic problems involve, in an abstract sense, a set of data objects and a set of relationships describing the data objects. The data objects and relationships constitute a knowledge base which is generally arranged as sets of facts and rules. A fact is a statement connecting a relationship with one or more data objects so that the statement is always interpreted as true. On the other hand, a rule is a statement which defines a relationship using other relationships, data objects, and/or facts.

A symbolic logic problem is usually stated in the form of one or more queries which are questions concerning relationships and data objects. The queries are answered by applying logical inference to the knowledge base of rules and facts. This inference process generates a set of assertions (inferred facts) from the knowledge base. The solution to the queries, therefore, becomes a set of conclusions in the form of data objects, which is inferred from the set of assertions so as to satisfy the queries.

B. PROLOG

Symbolic logic problems are relatively common. They arise in areas such as expert systems and other artificial intelligence applications. In recent years, the computer science language PROLOG has become a tool for solving these types of problem on electronic computers.[1] For example, two goals of fifth-generation computers are (1) to develop a machine capable of logical inference and data base operations and (2) to design a language based on PROLOG that would be suitable for inferring and representing knowledge.[2]

To solve a query, electronic PROLOG sequentially searches for the knowledge base for the appropriate rules and facts. This search process uses a flexible pattern-matching technique called unification which involves searching, matching, and backtracking through the knowledge base.[3,4] The performance of electronic PROLOG is limited by its use of serial searching and backtracking. PARALOG, an implementation of PROLOG which uses parallel unification, addresses this issue and is currently under investigation.[2]

C. Role of Optics

It is well known that 2-D parallel optical processors inherently perform high-speed pattern matching. Such systems should, therefore, be more efficient at searching than their serial electronic counterparts because the parallelism eliminates the need for backtracking through the knowledge base. Furthermore, since searching and pattern matching processors do not require high accuracy or large dynamic range, optical processors should in principle be well suited for symbolic logic processing.

We believe, however, that optical inference machines should be designed to be compatible with electronic computers. The goal should be to exploit the strengths of both systems so as to realize hybrid inference machines that are more efficient and versatile than either purely electronic or optical computers. For example, an optical inference machine could potentially be integrated into an electronic fifth-generation computer so that a hybrid machine capable of operating at speeds in excess of 10^9 logical inferences per second (LIPS) could be produced.

The authors are with MIT Department of Electrical Engineering & Computer Science, Cambridge, Massachusetts 02139.

Received 19 June 1985.
0003-6935/86/060940-08$02.00/0.

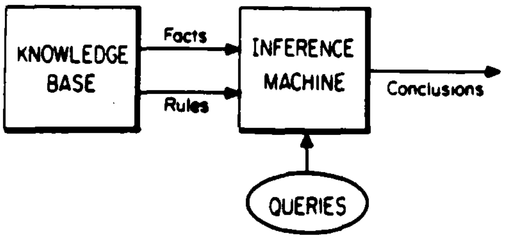


Fig. 1. General structure of an inference machine.

D. History

Previous work in optical symbolic processing was performed by several researchers in the late 1960s and early 1970s. Gabor,[5] Akahori and Sakurai,[6] Nakajima *et. al.*,[7] and Lohmann and Werlich[8] used holography as the basis for their processing techniques. Willshaw *et. al.*,[9] Willshaw and Longuet-Higgins,[10] and Gabor[5] approached the problem using associative network concepts. However, during the 1970s and early 1980s, the emphasis of research on optical computing systems shifted to numerical problems such as matrix–matrix multiplication,[11–13] array processing,[14] and solving sets of linear equations.[15]

More recently, there has been a resurgence of interest in the area of optical symbolic processing. Huang[16,17] has addressed the symbolic problem in a general sense, investigating algorithms and architectures for performing symbolic substitution optically in classical finite-state machines. Furthermore, Huang[18] and Fisher *et al.*[19] have recognized that there may be a possible role for optics in symbolic processors, particularly in solving certain classes of artificial intelligence problems. However, specific applications of optical computers to symbolic logic processing appear, until now, to have been unaddressed.

In this paper, the concepts associated with symbolic logic processors are introduced, and the general architecture of an optical machine capable of inferring logical conclusions from a set of facts and rules is discussed. The general system is approached from two different information flow patterns: rule-driven and query-driven flow. Two hybrid optical realizations for a query-driven inference machine are presented which use classical matched-filter logic and mapped-template logic, respectively. The intent here is to describe these systems from a conceptual point of view. Therefore, no attempt is made to address all the issues involved in realizing a practical system.

II. General Inference Machine Architecture

The general structure of an inference machine is shown in Fig. 1. It accepts as input a set of facts and a set of rules from the knowledge base and one or more queries. The output of the inference machine is a set of specific conclusions which are logically inferred from the facts and rules in response to the queries.

For example, a set of data objects could be a set of names of people. For illustrative purposes, let this set be denoted as

$$\mathbf{D} = \{\text{Karen, Beth, Peg, Liz, Sue, Jean, Ruth, Mike, Tom, Bill, Jim, Fred, Bob, Sam}\}. \quad (1)$$

A set of relationships for **D** might be the possible relationships between the people, such as marriage, mother, father, male, and female. Let this set of relationships be denoted by

$$\mathbf{R} = \{\textbf{married-to, mother-of, father-of, son-of, daughter-of, child-of, is-male, is-female}\}. \quad (2)$$

The data objects and relationships are linked as a collection of facts and rules which relate the elements of **D** and **R**. In this example, the facts could be defined as

Mike **is-male.**	Karen **is-female.**	
Tom **is-male.**	Beth **is-female.**	
Bill **is-male.**	Peg **is-female.**	
Jim **is-male.**	Liz **is-female.**	
Fred **is-male.**	Sue **is-female.**	
Bob **is-male.**	Jean **is-female.**	
Sam **is-male.**	Ruth **is-female.**	(3)
Mike **married-to** Karen.	Bob **father-of** Peg.	
Bob **married-to** Beth.	Bob **father-of** Tom.	
Jim **married-to** Liz.	Bob **father-of** Jean.	
	Jim **father-of** Ruth.	
	Fred **father-of** Bill.	

Using these facts, the remaining relationships in **R** may be defined as rules. For example,

$$\begin{array}{rll} X\ \textbf{mother-of}\ Y & \text{IF} & Z\ \textbf{married-to}\ X\ \text{AND} \\ & & Z\ \textbf{father-of}\ Y. \\ X\ \textbf{child-of}\ Y & \text{IF} & Y\ \textbf{mother-of}\ X\ \text{OR} \\ & & Y\ \textbf{father-of}\ X \\ X\ \textbf{son-of}\ Y & \text{IF} & X\ \textbf{child-of}\ Y\ \text{AND} \\ & & X\ \textbf{is-male,} \\ X\ \textbf{daughter-of}\ Y & \text{IF} & X\ \textbf{child-of}\ Y\ \text{AND} \\ & & X\ \textbf{is-female,} \end{array} \quad (4)$$

where X, Y, and Z are variables. The bodies of these rules (i.e., the part to the right of IF) consist of two conditions, each of which could be a fact or another rule. These conditions are then connected by the logical operators AND or OR. In general, a rule could have any number of conditions, and a condition could have a logical NOT operation performed on it. For example, the **daughter-of** rule could be modified to use the **son-of** rule by defining it with

$$\begin{array}{rll} X\ \textbf{daughter-of}\ Y & \text{IF} & X\ \textbf{child-of}\ Y\ \text{AND} \\ & & \text{NOT}\ X\ \textbf{son-of}\ Y. \end{array} \quad (5)$$

To satisfy a rule, there must be at least one data value for all variables for which all conditions are simultaneously true. In the **mother-of** rule, there must be at least one value each for X, Y, and Z so that Z is both married to X and the father of Y. Using the format of Eq. (4), additional relationships such as **sister-of** and **brother-of** are straightforward to define. Together, the facts in Eq. (3) and the rules in Eq. (4) form the knowledge base.

In general, a query into a knowledge base consists of a rule with at least one variable. For example, a possible query of this knowledge base could be "Who is the mother of Jean?", which can be expressed as

? **mother-of** Jean. (6)

where ? represents the desired unknown data object. From the knowledge base, only the assertion Beth **mother-of** Jean is true. Hence the conclusion of Eq. (6) is that the query is true when ? is the data object Beth.

Given a query and knowledge base, conclusions can be inferred using either inductive or deductive reasoning. In the inductive case, conclusions of a general nature are inferred by the application of specific queries to the knowledge base. The cardinality of the set of induced conclusions could in general be quite large, and, in principle, conclusions not representative of the knowledge base would be possible.

On the other hand, deductive reasoning produces specific conclusions from a set of general rules and facts, and the conclusions are always a subset of the knowledge base. For simplicity and practicality, we shall limit the allowed conclusions to the data objects within the knowledge base. Therefore, in this paper, we will consider only machines based on deductive reasoning.

Block diagrams for two general architectures of a deductive inference machine are shown in Figs. 2 and 3. Both systems have in common a knowledge base, controller, and inference filter. The functions of the controller are to (1) control the flow of information through the inference machine, (2) accept queries as input from the operator, and (3) transmit conclusions to the operator as output. The knowledge base stores all the data objects and relationships in the form of facts and rules. The role of the inference filter is to generate a set of all conclusions possible given a set of rules and facts from the knowledge base.

The system in Fig. 2 corresponds to a rule-driven inference machine, whereas that in Fig. 3 represents a query-driven inference machine. The systems are distinguished from each other by the methods they employ to infer the conclusions. In the rule-driven system, all possible assertions and facts from the knowledge base are generated *ab initio*, and thereafter the conclusions are derived from these inferences by application of the query. In contrast, the query-driven system first uses the query to select appropriate subsets of the rules and facts and then infers specific conclusions from these rules and facts.

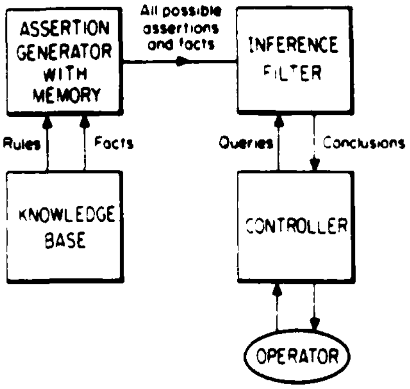


Fig. 2. Block diagram of a deductive rule-driven inference machine.

The rule-driven system of Fig. 2 approaches the ideal parallel system in that the assertion generator produces the facts and all possible assertions from the entire knowledge base by replacing all the rules with appropriate assertions. In the previous example, the **mother-of**, **child-of**, **son-of**, and **daughter-of** rules would lead to the assertions

Beth **mother-of** Peg.
Beth **mother-of** Tom.
Beth **mother-of** Jean.
Liz **mother-of** Ruth.

Tom **son-of** Bob.
Tom **son-of** Beth.
Bill **son-of** Fred. (7)

Peg **child-of** Bob.
Peg **child-of** Beth.
Tom **child-of** Bob.
Tom **child-of** Beth.
Jean **child-of** Bob.
Jean **child-of** Beth.
Ruth **child-of** Jim.
Ruth **child-of** Liz.
Bill **child-of** Fred.

Peg **daughter-of** Bob.
Peg **daughter-of** Beth.
Jean **daughter-of** Bob.
Jean **daughter-of** Beth.
Ruth **daughter-of** Jim.
Ruth **daughter-of** Liz.

Thus the output of the assertion generator would be the set of facts and assertions defined by Eqs. (3) and (7). Note that the knowledge base is not updated by the assertion generator and that the output produced by the assertion generator is computed only once.

As shown in Fig. 2, the assertion generator of the rule-driven machine transfers the entire set of facts and assertions to an inference filter whose function is to match the queries from the controller with the facts and assertions to determine the data objects which satisfy the queries. After it has determined the conclusions for the query, the inference filter transfers the conclusions to the controller for output to the operator.

In the example ? **mother-of** Jean, the inference filter would compare the facts and assertions defined by Eqs. (3) and (7) with the query given in Eq. (6). Realizing that ? is the desired variable, the filter would find a match between the query with the third assertion given in Eq. (7) to obtain the answer Beth. In this example, there was only one possible conclusion, but, in general, several data objects may satisfy a query.

In contrast, the query-driven system of Fig. 3 is a more sequential machine than the rule-driven system of Fig. 2. Given a query from the operator, the controller uses the rules associated with the query to select subsets of rules and facts from the knowledge base that are relevant to the query. In the example of Eq. (6), the **mother-of** rule is associated with the query. The controller would examine the **mother-of** rule as defined in the knowledge base and extract its condition relationships **married-to** and **father-of**.

Once it has obtained the necessary subsets of rules and facts, the controller transfers these subsets to the inference filter along with the known data objects from the query [Jean in Eq. (6)]. The inference filter then matches the rules with the known query data to infer the set of data objects which make the query true [Beth for Eq. (6)]. Finally, the inference filter sends the

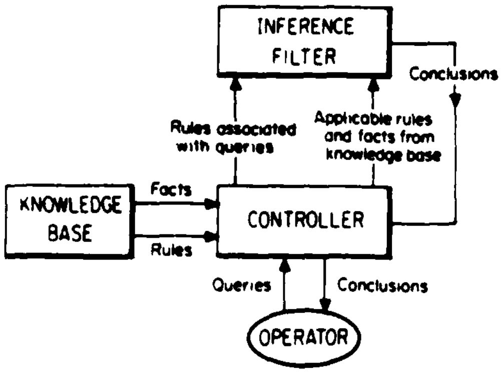


Fig. 3. Block diagram of a deductive query-driven inference machine.

conclusions back to the controller for output to the operator.

When consideration is given to implementation of an inference machine, the query-driven system may appear more attractive than the rule-driven system. This is because inferring the possible assertions and storing all the possible assertions and facts in the rule-driven system could be inefficient, expensive, and difficult to realize, particularly for rules which are recursively defined (i.e., when the rule has itself as a condition). Consequently, only query-driven systems are considered in the remaining sections.

III. Hybrid Optical Realizations

We shall confine our discussion to optical inference machines that complement the electronic computer. A complete system, therefore, will be hybrid in nature. This places design constraints on the input and output interfacing devices of the optical system. The optimum designs, therefore, are those that most effectively combine the individual strengths of optics and electronics. Two query-driven designs are described below, the first of which uses matched-filter logic in the inference filter, whereas the second is based on mapped-template logic.

In these systems, the parallelism and speed of optics are exploited to perform the functions of searching, matching, and logic. The role of the electronics is to perform information storage and retrieve and transfer data, rules, and operator queries to the optical processor. Thus in Fig. 3 the inference filter is the optical processor, while the controller and knowledge base constitute the electronic support system.

To implement these optical inference machines, three types of optical devices are required: (1) an input interfacing device which converts electrical signals to 2-D optical signals; (2) an optical logic device; and (3) an output interfacing device for transforming optical signals into electrical signals. The input interfacing device and optical logic device should exhibit at least short-term storage.

In the specific systems discussed below, the electrical-to-optical input device could be any 2-D electrically addressed spatial light modulator (E-SLM) which has short-term storage, such as the e-beam MSLM.[20] An example of an optical logic device which can perform 2-D logic with memory is the photo MSLM,[21,22] which is an optically addressed spatial light modulator (O-SLM). The logic operations that can be performed internally by the photo MSLM include AND, OR, NAND, NOR, XOR, and NOT. The optical-to-electrical output device is a 2-D photodetector array. To obtain good noise rejection and low error rates, digital optical signals (binary intensity levels) are assumed for all input and output signals in the optical processor.

A. Matched-Filter Optical Inference Machine

The general matched-filter optical inference machine employs analog pattern recognition techniques and parallel optical logic to apply a set of given rules to a set of facts to infer a set of logical conclusions to the queries. This method is similar to the optical correlograph system described by Willshaw and Longuet-Higgins.[10]

Figure 4 shows a specific implementation of a query-driven matched-filter hybrid optical inference machine. This machine consists of an electronic controller, two E-SLMs, two O-SLMs, and a photodetector array which is operated in a thresholding mode. In this and subsequent figures, it should be noted that (1) the input light to the O-SLM is absorbed within the device and is not transmitted, and (2) the readout light is reflected out of the device by an internal mirror.

In the matched-filter system of Fig. 4, the facts and rules are grouped in block form (subsets) and stored electronically in the controller for rapid retrieval and transfer to the optical system. The two E-SLMs, O-SLM 1, the lenses L_1, L_2, and L_3, and the photodetector array are arranged to form a classical VanderLugt matched-filter system.[23] Thus lens L_1 is one focal length away from the planes P_1 and P_2, lens L_2 is one focal length away from planes P_3 and P_4, and lens L_3 is one focal length away from planes P_3, P_5, and P_6. The multiplication of the Fourier transforms of the signals

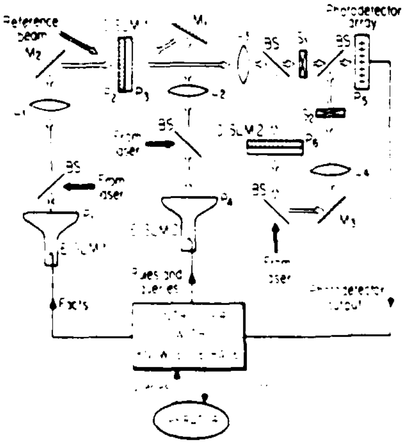


Fig. 4. Matched-filter optical inference machine.

to be matched is performed in O-SLM 1, and the matched-filter output is recorded on the photodetector array (shutter S_1 open, S_2 closed). The photodetector then transfers its output to the controller.

If the query dictates that several rules must be applied to the facts in succession, the resulting matched-filter outputs can be combined by using the optical logic capabilities of O-SLM 2. With S_1 closed and S_2 open, the logic output of O-SLM 2 can be imaged onto the photodetector array using lens L_4 and the photodetector output fed back to the controller. This ability permits rules to be applied as many times as necessary to various subsets of facts to generate the logical conclusions.

When operating the matched-filter optical inference machine, the operator queries the system through the electronic controller. In response, the controller writes the applicable subsets of facts onto E-SLM 1 and the applicable subset of rules onto E-SLM 2. This information is coded as a set of predetermined 2-D binary-level patterns. In the query example of Eq. (6), the **mother-of** rule and the complete set of facts in Eq. (3) would be the applicable sets.

The controller then activates O-SLM 1 which holographically records the Fourier transform of the facts as formed by lens L_1. The rules are similarly transformed by lens L_2, and this transform is used to read out O-SLM 1 via mirror M_1 as shown in Fig. 4. The output of O-SLM 1 is transformed by lens L_3 to form the matched-filter output on the photodetector array. This output consists of a set of focused spots of light which indicates the positions of the matches. These signals are then stored in O-SLM 2 and/or fed back to the controller, which then uses this input to select the possible conclusions from the set of facts.

Several options exist at this point, depending on the nature of the query being solved. For example, the controller could now load another part of the query into E-SLM 2, perform a second matched-filtering operation, and with S, closed and S_2 open perform a logical AND (with O-SLM 2) of the second correlation and the first which is already stored in O-SLM 2. The output of O-SLM 2 would then be read out onto the photodetector array. Thus the matched-filter inference machine is capable of sequentially performing all combinations of 2-D optical pattern correlations and binary level logic operations on patterns representing the data objects, rules, facts, and queries.

To solve the ? **mother-of** Jean query in Eq. (6), this system would first examine the query for the specified data objects (Jean in this case) and would then treat the **mother-of** rule as if its variables were replaced by the appropriate data objects. In this case, the effective **mother-of** rule would become

$$? \textbf{ mother-of } \text{Jean IF } Z \textbf{ married to } ? \text{ AND } Z \textbf{ father-of } \text{Jean.} \quad (8)$$

Comparing Eq. (8) with the original **mother-of** rule as defined in Eq. (4), the variables X and Y have been replaced with the desired unknown symbol ? and the data object Jean, respectively. Since the **mother-of** rule has two conditions, the controller has to invoke two matched-filtering operations.

The order in which the conditions are satisfied does not matter since all of them must be true for the query to be satisfied. Since the second condition has the data object Jean as a constraint, the first matched-filtering operation matches the **father-of** facts (placed on E-SLM 1) with the data object Jean (placed on E-SLM 2). The output of the matched-filter is then a representation of all facts associated with the condition **father-of** Jean. In this case, there is only one fact associated with this condition, Bob **father-of** Jean. The controller then retrieves the father's name Bob and matches the condition Bob **married-to** with the set of facts. The second matched-filter output points to the fact Bob **married-to** Beth. Finally, the controller simply associates the conclusion Beth with ? and returns the conclusion to the operator.

In the case where there are several matches, it is possible for the controller to match all the resulting conclusions with the next condition for full parallelism. Furthermore, if no match is made (i.e., no spots of light above threshold on the photodetector array), the condition cannot be satisfied, making the query false.

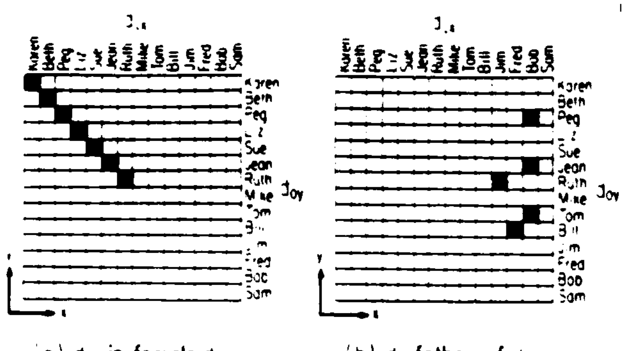
The block electronic storage scheme suggested here is not the most efficient means of storing the rules and the facts because a single data object may be associated with several different facts. However, because electronic storage is relatively inexpensive, block-form storage does not appear to be inappropriate for the initial investigations of these machines.

Since data objects are not expected to change often, partitioning the knowledge base into blocks will generally not have to be done frequently. The advantage of block electronic storage is that it not only reduces the data acquisition and retrieval time but also eliminates the need to transfer the entire knowledge base to the spatial light modulators which currently have only modest space–bandwidth products.

B. Mapped-Template Optical Inference Machine

In the mapped-template optical inference machine, mapping templates are used to store the relationships between the data objects and are thus defined by the facts. Conclusions are inferred to queries by applying these mapping templates to the data objects in the order prescribed by the rules. This usage of mapping templates is similar to the associative nets described by Willshaw and Longuet-Higgins.[10]

Using the example defined by Eqs. (1)–(6), the relations **is-male, is-female, married-to,** and **father-of** from the facts in Eq. (3) would map an input set from **D**, the set of data objects defined in Eq. (1), to an output set, also from **D**. Let $\mathbf{D}_i$ and $\mathbf{D}_o$ represent the input and output sets of data objects. Furthermore, let the data objects in the mth position of $\mathbf{D}_i$ and $\mathbf{D}_o$ be denoted by d_{im} and d_{om}. Using the data set **D** for $\mathbf{D}_i$ and $\mathbf{D}_o$ as defined in Eq. (1), the mapping templates corresponding to the **is-female** and **father-of** facts as defined in Eq. (3) are shown in Fig. 5 with the elements



(a) d_{ix} is-female d_{oy} (b) d_{ix} father-of d_{oy}

Fig. 5. Mapping templates for (a) the **is-female** facts and (b) the **father-of** facts for the entire set of data objects in Eq. (1).

of the input set $\mathbf{D}_i(d_{ix})$ along the columns (x axis) and the elements of the output set $\mathbf{D}_o(d_{oy})$ along the rows (y axis) of the templates.

The mapping templates are binary masks consisting of transparent squares (logical 1 and shown as black squares in Fig. 5) on an opaque background (logical 0 and shown as white in Fig. 5). The interpretation of these templates is as follows: A transparent square in the (x, y) position of, say, **father-of** indicates the fact

$$d_{ix}\ \textbf{father-of}\ d_{oy}. \tag{9}$$

Given these two templates, the mapping templates for **is-male** and **married-to** are straightforward to generate.

Note that the mapping between $\mathbf{D}_i$ and $\mathbf{D}_o$ is not necessarily one-to-one. However, a mapping template is reciprocal in that if the right-hand side of Eq. (9) is specified instead of the left, the relationships for the left-hand side may be inferred from the template.

Alternatively, to limit the size of the mapping template and conserve space, **D** could be subdivided into subsets whose data objects are related in some way. Considering the facts in Eq. (3), it is reasonable to split **D** into a set of males and a set of females denoted by

$$\begin{aligned}\mathbf{D}_m &= \{\text{Mike, Tom, Bill, Jim, Fred, Bob, Sam}\}\\ \mathbf{D}_f &= \{\text{Karen, Beth, Peg, Liz, Sue, Jean, Ruth}\},\end{aligned} \tag{10}$$

where $\mathbf{D}_m$ and $\mathbf{D}_f$ represent the male and female sets, respectively. With these subsets, the relationships **is-male** and **is-female** would no longer be needed.

With the data set partitioned, the mapping templates for the factual relationships between the elements of $\mathbf{D}_m$ and $\mathbf{D}_f$ would simply be the corresponding regions in the original full-size mapping templates in Fig. 5. For the relation **is-female** and the data set $\mathbf{D}_m$, the template would always be opaque.

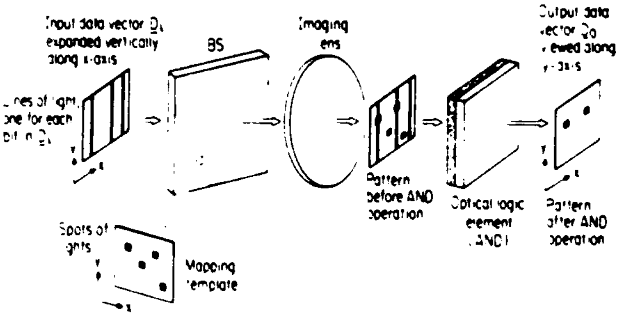
To perform logical inferring, the mapping-template concept is implemented as illustrated in Fig. 6. Given an input vector $\mathbf{D}_i$, the associated output vector $\mathbf{D}_o$ for a particular mapping template is found by first vertically expanding $\mathbf{D}_i$ along the y axis so it forms an array, each row of which equals $\mathbf{D}_i$, as shown in Fig. 6. This expanded form of $\mathbf{D}_i$ is then optically overlaid with the mapping template using imaging optics and a 2-D logical AND operation is performed. The resulting output, when viewed along the rows, corresponds to the output vector $\mathbf{D}_o$.

To perform the reciprocal operation of the mapping template, the input vector would be expanded horizontally and logically ANDed with the mapping template. The output vector would then be taken looking down the columns.

Depending on the mapping template, it is possible for multiple inputs in $\mathbf{D}_i$ to produce the same output element in $\mathbf{D}_o$. For this reason, a 2-D output photodetector array is used for establishing the exact input-to-output correspondence, should this be needed in solving the query.

A hybrid optical inference machine which implements mapped-template logic is shown in Fig. 7. It consists of an electronic controller, two E-SLMs, two O-SLMs, and a 2-D photodetector array. Like the matched-filter optical inference machine, the controller in this system electronically stores the knowledge base and controls the SLMs and the shutter. The modulator O-SLM 1 is operated in the logic mode and usually performs the AND operation, while O-SLM 2 is used as a 2-D memory unit to allow further processing of the outputs, and is optional.

When the controller is given a query by the operator, a vertical line is written on E-SLM 1 at the location of



Input data vector D_i expanded vertically along x-axis
BS
Imaging lens
Output data vector D_o viewed along y-axis
Lines of light, one for each bit in D_i
Pattern before AND operation
Optical logic element (AND)
Pattern after AND operation
Spots of lights
Mapping template
x
y

Fig. 6. Conceptual implementation of mapped-template logic.

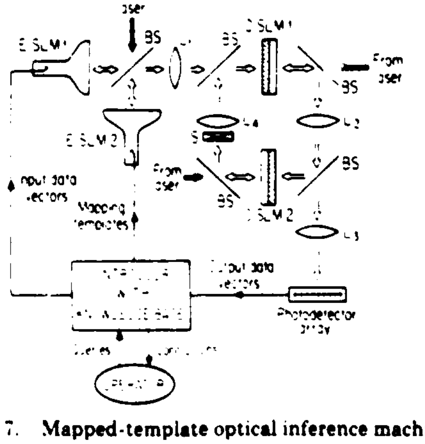


Fig. 7. Mapped-template optical inference machine.

the known data objects in $\mathbf{D}_i$. Then the controller writes the mapping template corresponding to the rule (or first condition) associated with the query onto E-SLM 2. The outputs of both E-SLMs are imaged onto O-SLM 1 with lens L_1. The logical AND of the two inputs is formed in O-SLM 1 and imaged onto the photodetector array by lenses L_2 and L_3. If desired, the output could also be imaged onto O-SLM 2 by lens L_2 and latched. The stored output in O-SLM 2 could then be imaged via lens L_4 back into O-SLM 1 by opening shutter S should further processing be necessary.

The output of the photodetector array is fed back to the controller where the inferred data objects in $\mathbf{D}_o$ which satisfy the current mapping rule are determined. Further mapping templates are then applied by the controller as determined by the query and rules.

Operation of this optical inference machine can be demonstrated for the ? **mother-of** Jean query in Eq. (6). As with the matched-filter machine, the mapped-template system considers the effective form of the **mother-of** rule given the data object Jean as specified in Eq. (8). The controller first uses the mapping rule template for **father-of** as shown in Fig. 5 and the input vector corresponding to Jean, which is, from Eq. (1), [0 0 0 0 0 0 1 0 0 0 0 0 0 0]. Since Jean is specified on the output side of **father-of**, the input vector is expanded horizontally rather than vertically on E-SLM 1. Scanning the rows of the output array produces the output vector [0 0 0 0 0 0 0 0 0 0 0 0 1 0] which corresponds to Bob.

The controller then feeds this output vector back to E-SLM 1 as the input vector for the **married-to** mapping template. Since this input is on the right side of the **married-to** rule, the vector is expanded vertically on E-SLM 1. The inference operation is repeated with the **married-to** mapping template on E-SLM 2, producing the output vector (view along the columns) [0 1 0 0 0 0 0 0 0 0 0 0 0 0], which indicates the conclusion Beth.

Since multiple outputs for the same data object could be generated, viewing the rows or columns of the output array could lead to an integral multiple of a single light beam intensity. In this case, the photodetector output is electronically clipped to the single light beam level if the photodetector output is to be fed back to E-SLM 1 as input via the controller.

If the optional optical feedback loop is not used, there is a possible modification to the system in Fig. 6 which will simplify the device requirements. Instead of performing the logical AND operation in O-SLM 1, the output of E-SLM 1 (the expanded input data vector) could be used to read out the mapping template in E-SLM 2, thus eliminating the need for a 2-D optical logic device. However, the advantage of having O-SLM 1 is that (1) it can conveniently perform the logical NOT operation on a condition, and (2) the processed patterns are automatically latched into O-SLM 1. This allows the controller to begin setting up the next mapping template while it simultaneously reads the photodetector array, thus providing some degree of concurrent operation.

Further possibilities for increasing processing speed are to place multiple mapping templates which are spatially separated from each other on E-SLM 2. The input data vectors on E-SLM 1 would have to be repositioned accordingly. However, multiple inferences could then be made in parallel.

IV. Concluding Remarks

Basic architectures for a hybrid optical machine capable of solving symbolic logic problems have been discussed in general terms. This inference machine was considered from both a rule-driven and query-driven approach. Two hybrid optical designs of a query-driven inference machine were described which used matched-filter logic and mapped-template logic.

In comparing the two designs, the mapped-template system should be less demanding on the spatial resolution characteristics of the spatial light modulators and should be easier to implement than the matched-filter machine. Furthermore, the mapped-template system should have better noise performance since there is no analog processing in this system. That is, all optical signals remain encoded as binary intensity levels in the mapped-template system, whereas the matched-filter system must contend with the noise from the analog matched-filtering process, even though binary intensity input and output patterns are used.

Although two hybrid architectures have been presented, other equally effective system designs are possible. Given the growing interest in integrating symbolic logic processing into the computer of the future, the idea of downloading the inference operations of scanning, searching, and matching to a parallel optical processor merits continued investigation.

This work was supported in part by the Air Force Office of Scientific Research under grant AFOSR-84-0358.

References

1. D. S. Nau. "Expert Computer Systems." IEEE Comput. 63 (Feb. 1983).
2. T. Moto-oka and H. S. Stone. "Fifth-Generation Computer Systems: A Japanese Project." IEEE Comput. 6 (Mar. 1984).
3. K. L. Clark and F. G. McCabe. *Micro-Prolog Programming in Logic* (Prentice-Hall. Englewood Cliffs. NJ. 1984).
4. W. F. Clocksin and C. S. Mellish. *Programming in Prolog* (Springer-Verlag. 1981) New York.
5. D. Gabor. "Associative Holographic Memories." IBM J. Res. Dev. 156 (1969).
6. H. Akahori and K. Sakurai. "Information Search Using Holography." Appl. Opt. 11, 413 (1972).
7. M. Nakajima. T. Morikawa. and K. Sakurai. "Automatic Character Reading Using a Holographic Data Processing Technique." Appl. Opt. 11, 362 (1972).
8. A. W. Lohmann and H. W. Werlich. "Holographic Production of Spatial Filters for Code Translation and Image Restoration." Phys. Lett. A 25, 570 (1967).
9. D. J. Willshaw. O. P. Buneman. and H. C. Longuet-Higgins. "Non-Holographic Associative Memory." Nature London **222**, 960 (1969).

10. D. J. Willshaw and H. C. Longuet-Higgins, "Associative Memory Models," Machine Intell. 5, 351 (1970).

11. R. A. Athale, W. C. Collins, and P. D. Stilwell, "High Accuracy Matrix Multiplication With Outer Product Optical Processor," Appl. Opt. 22, 368 (1983).

12. R. P. Bocker, "Optical Digital RUBIC (Rapid Unbiased Bipolar Incoherent Calculator) Cube Processor," Opt. Eng. 23, 26 (1984).

13. Y. Z. Liang and H. K. Liu, "Optical Matrix-Matrix Multiplication Method Demonstrated by the Use of a Multifocus Hololens," Opt. Lett. 9, 322 (1984).

14. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz, "Optical Implementation of Systolic Array Processing," Opt. Commun. 40, 86 (1981).

15. M. Carlotto and D. Casasent, "Microprocessor-Based Fiber-Optic Iterative Optical Processor," Appl. Opt. 21, 147 (1982).

16. A. Huang, "Parallel Algorithms for Optical Digital Computers," Proc. Soc. Photo-Opt. Instrum. Eng. 422, 13 (1983).

17. K. Brenner and A. Huang, "An Optical Processor Based on Symbolic Substitution," in *Technical Topical Meeting on Optical Digest, Computing* (Optical Society of America, Washington, D.C., 1985), paper WA4.

18. A. Huang, "Why Use the Parallelism of Optics?," in *Technical Digest, Topical Meeting on Optical Computing* (Optical Society of America, Washington, D.C., 1985), paper WA2.

19. A. D. Fisher, C. L. Giles, and J. N. Lee, "An Adaptive, Associative Optical Computing Element," in *Technical Digest, Topical Meeting on Optical Computing* (Optical Society of America, Washington, D.C., 1985), paper WB4.

20. A. Schwartz, X. Y. Wang, and C. Warde, "Electron-Beam-Addressed Microchannel Spatial Light Modulator," Opt. Eng. 24, 119 (1985).

21. C. Warde, A. M. Weiss, A. D. Fisher, and J. I. Thackara, "Optical Information Processing Characteristics of the Microchannel Spatial Light Modulator," Appl. Opt. 20, 2066 (1981).

22. C. Warde and J. I. Thackara, "Oblique-Cut $LiNbO_3$ Microchannel Spatial Light Modulator," Opt. Lett. 7, 344 (1982).

23. A. B. VanderLugt, "Signal Detection by Complex Spatial Filtring," IEEE Trans. Inf. Theory IT-10, 2 (1964).

PUBLICATIONS RESULTING FROM AFOSR SPONSORSHIP

1. J. Kottas and C. Warde, "Versatile Optical Circuits," in the conference Digest of the 13th Congress of the International Commission for Optics, Sapporo, Japan, 1984, pp160-163.

2. C. Warde and J. Kottas, Versatile Two-Dimensional Optical Circuit Arrays", in the Technical Digest of the Winter 1985 Conference on Optical Computing, Incline, Village, Nevada, Paper TuC4-3.

3. C. Warde and J. Kottas, "Hybrid Optical Inference Machines: Architectural Considerations" Appl. Opt. 26, 940-947, 1986.

M.S. THESES

J. Kottas, "Optical Pattern Processors for Matrix Multiplication and Symbolic Inference, 1986

LIST OF PERSONNEL

Faculty and Staff

Professor Cardinal Warde

Graduate Students

James Kottas
Doyle Temple
R. Scott Hathcock
Suzanne Lau

Undergraduate Students

Phillip Mak
Ling-Yi Liu

Visting Scientist

H. Lamela Rivera
Kuang-Yi Huang